Master thesis

MOT TU Delft

R. Meijer

ETA prediction
Predicting the ETA of a container vessel based on route identification using AIS data

ETA prediction

Predicting the ETA of a container vessel based on route identification using AIS data

by

R. Meijer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 22, 2017 at 15:30 PM.

Student number:	4420004	
Project duration:	February 1, 2017 – July 1, 2017	
Thesis committee:	Prof. dr. Y. Tan,	TU Delft, Chair
	Dr. ing. V. E. Scholten,	TU Delft, First supervisor
	Dr. A. M. Herdeiro Teixeira,	TU Delft, Second supervisor
	MSc. F. Fanitabasi,	TU Delft, Third supervisor

TU Delft

Abstract

Container vessels arriving in a port before or after their scheduled time can cause problems in the container terminal planning and planning of hinterland transportation. This in turn leads to an increase of the costs in the supply chain. Vessels communicate their Estimated Time of Arrival via Automatic Idetification System(AIS) data to the port. This arrival time is estimated by the crew of the vessel and manually inputted into the AIS. In this research a proof of concept is shown that the Estimated Time of Arrival (ETA) prediction of container vessels can be improved. Vessels en route to the Port of Rotterdam are used as a case study. Different frameworks and algorithms are introduced to improve the data quality of AIS messages, to identify a set of possible routes and to do predictions based on the set of possible routes. It is possible to do predictions based on pre-processed AIS messages and a set of possible routes that perform at the same level as the best guess of a vessel's crew.

Contents

List of Algorithms

Acknowledgements

I would like to thank Prof. Tan, the chair of my research committee , and Dr. Herdeiro Teixeira and Dr. Scholten, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to extend my deep gratitude to Farzam Fanitabasi, for his advice and assistance in keeping my progress on schedule.

Finally, I wish to thank my parents and friends for their support and encouragement throughout my study.

1 Introduction

In this thesis a framework is presented to predict the Estimated Time of Arrival of container vessels. The presented framework is able to provide an Estimated Time of Arrival that is as good as the best possible guess provided by the captain of a vessel. Data is needed to base the predictions on. AIS messages are a commonly used data source in the maritime domain and has a lot of advantages. However the data quality of AIS messages is affected by multiple issues. So assessing and improving the data quality of AIS messages is also covered by the framework. The problem statement, objectives, relevance and research questions are introduced in this chapter. Also the research approach, deliverables, the expected outcome and structure of the thesis are discussed.

1.1. Problem statement & Objectives

In the transportation domain, transportation via containers is one of the most important ways of transportation. Most containers are shipped via container vessels that travel the world visiting all kinds of ports[76]. During these voyages a lot can happen what influences the travel times of these vessels. The arrival and knowledge about the arrival times of these vessels in a port are key for the planning in the terminal and of the remaining transport. However these arrival times are now estimated by a captain of a vessel and hardly accurate. This influences the processes in a container terminal and leads to increased costs in the supply chain. An elaborate discussion is presented in Chapter 2.

Previous research have shown that machine learning can lead to more accurate predictions for vessels on a direct route to a port[58]. However this is a rare occurrence in maritime transport and therefore the research is expanded to all vessels en route to a port. Every route has different characteristics and therefore the hypothesis is drawn up that it is necessary to incorporate the routes in a prediction to get accurate predictions.

One of the main problems in a containerized supply chain for ports is the uncertainty over arrival times of container vessels at the sea port[15, 58, 81]. This uncertainty impacts the planning of activities at the container terminal and the planning of hinterland transportation. Due to the uncertainties in the planning the cost of the services in the supply chain increase and thus the total cost of shipping goods[91, 92].

The communication of the ETA of a vessel is very limited since it is only mandatory 24 hours in advance and even after this time period the ETA is subject to a lot of changes due to unforeseen events. An accurate ETA prediction is needed by the container terminal 2 to 3 days in advance and by the forwarder and hinterland transportation carriers even 7 days in advance for their planning activities[58]. So the sub-objective is set to design a framework that provided predictions at least 7 days in advance.

With a reduction in the uncertainty regarding the ETA of a container vessel, the cost in the supply chain decrease as a results[91, 92]. Because a reduction in uncertainty leads to a reduction in changes in the planning, which in turn reduces waiting times, workload peaks and usage of ad-hoc trucks for example. All these reductions lead to less cost for one or multiple stakeholders in the supply chain. Therefore reducing the uncertainty of the arrival time of a container vessel is one of the sub-objectives of the research.

Predictions are improved by the use of AIS messages[58]. However AIS messages are error prone[12, 20]. So another sub-objective is considered to be assessing and improving the data quality of AIS messages.

The sub-objectives are set as follows:

- Predict ETA for all vessels en route to a port
- Incorporate routes into ETA predictions
- Provide predictions at least seven days in advance
- Reduce the uncertainty of arrival times
- Asses and improve the data quality of AIS messages

The main objective of this research is set by combining the different sub-objectives. To reduce the uncertainty of arrival times at least seven days in advance it is necessary to predict the ETA for all vessels en route to a port. The predictions are less uncertain when the accuracy is improved. To improve the accuracy, the incorporation of routes into the predictions is needed and furthermore data with a decent quality is needed. So the quality of AIS messages needs to be assessed and improved. So the objective is to design a methodology that improves the data quality of AIS messages and predicts the ETA of container vessels en route to a port at least seven days in advance. Since it is not feasible to construct a tool for every port, the Port of Rotterdam is used as a case study.

1.2. Scientific relevance

Research into the ETA of container vessels is very limited. To the knowledge of the writers of this thesis only Fancello et al.[13] and Parolas[58] have researched this topic. Their research is discussed in the literature review. Since the focus of the research of Fancello et al. is vague and the research of Parolas is focussed on a specific route, situation and time-span, the research in this field is expanded by this research. The literature review is used to identify a knowledge gap and identify that vessels that visit other ports prior to a certain destination have not yet been researched upon. Since this is common practice in maritime transport, it is necessary to create an algorithm that predicts the ETA of a vessel possibly visiting ports before the Port of Rotterdam.

To predict the ETA of a vessel that visits multiple ports, knowledge is needed about the route of that vessel. However the route is not communicated by a vessel to a port. This is a practical problem that needs a solution. Every port has its own characteristics, e.g. number of quay cranes, that influence the time a vessel spends in the port. So the number of ports and which ports are visited influence the time a vessel needs to arrive in the Port of Rotterdam. If every route is considered equal, the average ETA of all routes is predicted. This may improve the ETA prediction but also a huge variance is predicted since the travel time between a vessel with a direct line to Rotterdam and a vessel with for example three stops before Rotterdam can differ by days. In order to overcome these problems route identification is used to identify a possible set of routes for the vessels. It is hypothesized that with the incorporation more accurate predictions with a low variance are produced.

So in order to predict the ETA, the possible routes of a vessel need to be identified. In the research of Pallotta et al.[57], Nevell[51] and Lane et al.[35] the current location of a vessel is used to predict the possible routes. Pallotta et al. introduce the TREAD method that predicts the next possible destinations of a vessel. Nevell constructs a network and uses this network in combination with Bayes theorem to predict the possible destinations of a vessel[51]. Lane et al. expand on this research and introduce Hidden Markov Models to characterize the sequence of port arrivals[35]. Since vessels tend to visit the same ports in the same sequence[35], historical AIS data can be used to identify the routes that are used. When these routes are known, prediction models can be constructed for each route. As in the TREAD method from Pallotta et al.[57] vessels are observed in a certain area and their stops and next destination are tracked. This information can be used to decide which models to use to predict the ETA of container vessels. For instance vessels start in Felixstowe, England. Some vessels travel directly to the Port of Rotterdam, while others travel to the Port of Rotterdam via Hamburg. Since the vessel that travels via another destination covers a larger distance and also need time in the port to load and offload containers, they will need different prediction models. Therefore incorporating route identification is hypothesized to provide more accurate ETA predictions. A more elaborate explanation regarding route identification is given in Chapter 6. Machine learning techniques for ETA prediction are combined with route identification based on the methods of Lane et al.[35] and Pallotta et al.[57].

So in general this research is the first that predicts the ETA for vessels possibly visiting multiple ports over a multi-day time-span using historical data. Proven methods for ETA prediction are used and are expanded with the incorporation of route identification, something that has not yet been researched upon. In order to be able to predict the ETA, data is needed that meets certain standards. A lot of research has been conducted into possible issues with data while using AIS. But nobody designed a framework to accommodate these issues. So steps are included to improve data quality of AIS messages in the framework.

In summary the contributions of this research are as follows:

- Novel framework to improve the data quality of AIS messages
- Incorporating route identification into ETA prediction
- Recommendations to improve AIS messages

1.3. Practical relevance

In a supply chain, multiple stakeholders are involved. Providing abundance of information while improving the accuracy will have benefits for all involved stakeholders[91, 92]. But these benefits are different for every stakeholder. In this section the implications of wrong ETA's are shortly discussed per stakeholder and how more accurate predictions can alleviate these implications. A more elaborate discussion can be found in Chapter 8.

Terminal operators are responsible for the planning in the terminals. The planning in the terminal is based on the ETA of vessels. When wrong ETA's are communicated the planning is flawed and need to be adjusted ad-hoc, for instance when a vessel is late another vessel may already be occupying the allocated berth, so a new berth needs to be assigned ad-hoc. These changes lead to extra costs and cause (longer) waiting times for vessels. When the framework is able to predict the ETA for every vessel more accurate, the planning requires less (profound) changes.

Forwarders and hinterland carriers transport containers from the terminal to the hinterland. Forwarders buy slots at the hinterland carriers based on the predicted arrival times of containers. However due to the wrong information provided the amount of slots is often misaligned with the required amount of slots. So forwarders pay for unused slots or need to buy extra slots with expensive ad-hoc trucks. With more accurate information the amount of reserved slots will be closer aligned to the required amount of slots and thus the costs will decrease.

Ports are responsible for the safe passage of vessel from the sea to the terminal. For instance they make the planning of tugboats. The required amount of tugboats is based on the ETA's of a vessel. When more vessels arrive at a certain time then planned, this creates waiting times at sea because not enough tugboats are available to guide a vessel to its terminal. With more accurate ETA information the planning of these tugboats can be optimized. Furthermore because the tool is able to reduce waiting times and costs it will also improve the competitive position of a port.

Shipping lines are not directly affected by wrong ETA's since is does not influence their process. However the waiting times that are a consequence of wrong ETA's are affecting them. Because of these waiting times it might be that shipping lines miss their deadline and thus are obliged to pay a fine. Furthermore their personnel needs to be payed salary during these waiting times. Providing accurate ETA's will lead to shorter waiting times and thus less fines and salary to be paid by the shipping lines. Furthermore if a shipping line has knowledge on how many vessels are in a port, they can adjust their price in the negotiations.

1.4. Research question and approach

The objectives as discussed in Section 1.1 are translated into the research question:

How to improve the AIS-based ETA predictions of vessel en route to a port by leveraging route identification?

The research starts with a literature review. This review is used to show that more information in a supply chain is able to reduce the cost in a supply chain, identify possible data sources and the advantages and disadvantages related to the data source. After this literature review the methodology is introduced. The research is continued by assessing the data quality of AIS data. This assessment is used to improve the data quality of the received AIS data. The AIS data is pre-processed to make it ready for route identification and ETA prediction. When the data quality is improved by processing the data, route identification and how to incorporate this into predicting the ETA of a container vessel in a port is demonstrated. Finally the implications of the framework are discussed from the perspective of the stakeholders.

1.5. Deliverables and expected outcome

To answer the research question a framework is produced that assesses and improves the data quality of a dataset containing AIS messages and predicts the ETA of container vessels at a port based on a set of possible routes. Furthermore a report is produced that presents the research. It is expected that the research shows that using machine learning technologies for predicting the ETA in combination with route identification reduces the error in ETA prediction.

1.6. Document structure

The remainder of this theses is structured as follows. Background information on containerized supply chains is provided and used to identify wrong ETA's as a problem. The results of the literature review are presented in Chapter 3. Following the literature review, the methodology is presented in Chapter 4. The different steps identified in the methodology are used to conduct the research. An assessment of the data quality of AIS messages is made in Chapter 5. From this assessment pre-processing steps are identified in the same chapter to improve the data quality of AIS messages. The pre-processed data is used in Chapter 6 to identify a set of possible routes for a vessel based on AIS messages. The final steps of the framework are discussed in Chapter 7 where a prediction method is chosen, the model is trained and predictions for the ETA of a vessel are made. The possible benefits for the stakeholders are discussed in Chapter 8, in addition to this discussion the power of the stakeholders to affect the quality of AIS messages is discussed. Problems encountered during the research and their effect on the outcome are discussed in Chapter 9 in addition possible improvements to AIS and the methodology are discussed. The overall conclusion of the research is presented in the final chapter.

2

Background Information

In this chapter some background information is provided on containerized maritime transport. If you are not familiar with the domain reading this chapter is advised. Otherwise this chapter can be skipped.

2.1. Maritime transport

Nowadays we live in a globalized economy, all kinds of goods are produced all over the world and transported to every corner of the world. 80% of this trade is done via maritime transport[76]. Maritime transport is the major mode of transport due to its unmatched capacity, its ability to cover large distances, its low costs and its environment-friendly nature[1, 63, 87]. Maritime transportation routes span across hemispheres, transporting all kinds of materials, e.g. raw materials, parts and finished goods. This makes maritime transportation one of the most globalized industries[63].

Maritime freight transport consist of two main components. The first component consists of the different transportation modes which have a flexible nature regarding their spatial allocation. Shipping lines have the ability to choose their routes, the frequency and level of service. The second component consists of the ports who have a fixed position and capacity that has financial consequences if this capacity is unused. Ports are the crucial location where the maritime and land traffic converges in the globalized economy of today[63]. An enormous amount of ports are located around the world, however the top 20 of busiest ports accounts for almost half of the container throughput worldwide[76]. These ports cannot be easily neglected in shipping and therefore make up the global geography of trade and flows[63].

Finding the optimum where the capacity of terminals and vessels is matched is a huge challenge. One of the solutions to this challenge is that shipping lines invest directly in terminals to ensure hinterland access. These investments are necessitated by the steadily growing volume of maritime freight transport and increasing complexity due to an increasing amount of origins, destinations and supply chains[63]. However these increasing investments resulting in new capacity on shipping lines have also caused a downturn in the maritime freight transportation market, except for tankers. Because the demand for capacity is weakened but the supply in capacity has increased, shippers experienced historic low levels of freight rates and low earnings[76]. This new environment leads to an increasing importance of reliable and timely deliveries[63].

International freight transportation consists of two types of cargo, bulk and break-bulk. Bulk cargo is cargo such as ores, coal, grain and oils. Homogeneous products that have no packaging. Break-bulk cargo is also known as general cargo. This is cargo with all kinds of different shapes and sizes that are packaged together in bags, crates etcetera. However most general cargo is shipped in containers[63]. The first containers were shipped around the 1960s and have caused a huge transformation in the shipping industry[18, 63]. The containerization of the general cargo made it possible to ship the general cargo in containers of standard size, making it more easy to load and unload general cargo. This shortened the time it took to load and unload a vessel considerably. Another improvement was the ability to use those standard containers with multiple transport modes[63], so a container loaded on a vessel can be unloaded in a terminal and loaded onto a barge, train or truck for further transportation[18]. This is called intermodal transport. Due to these changes the entire supply chain is now seen as a whole and not as a series of stages anymore[63].

Since shipment via containers is such an important way of transportation an overview is provided of containerized maritime transportation in the next section.

2.2. Containerized supply chains

80% of the global merchandize is transported by sea and handled at ports, roughly 70% of maritime transportation is done with the help of containers which facilitate seamless transfer between multiple modes of transportation[76]. In this section containerized supply chains, the logistics network and the stakeholders in the supply chain are discussed. This discussion is used to identify possible problems for the research.

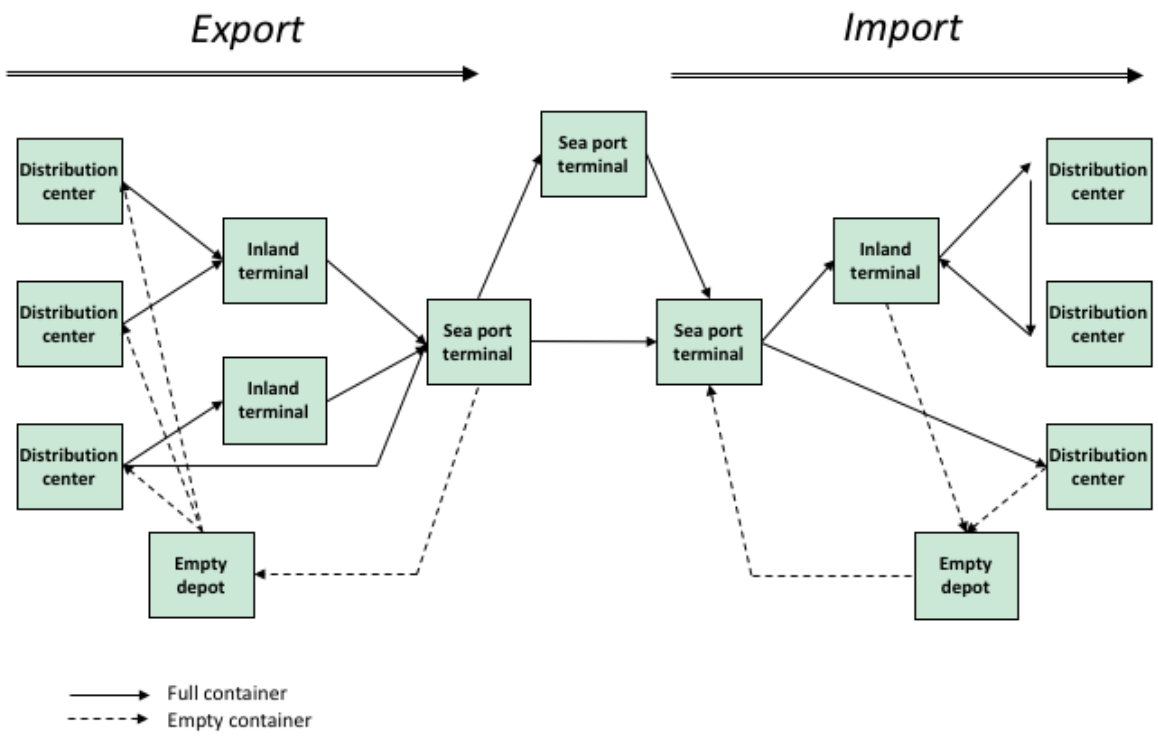


Figure 2.1: Overview of the logistics network of a containerized supply chain. Image from Zuidwijk[90]

2.2.1. Logistics network

Figure 2.1 shows an overview of the logistics network in a containerized supply chain. As presented in Figure 2.1 the sea port terminals are in the center of the supply chain connecting the export and import sides of the supply chain. They also connect the land-side and seaside of the supply chain[58, 90]. This makes the sea port terminal one of the most important parts of the logistics network and a smooth and efficient operation of the terminal is of vital importance for the performance of the entire supply chain[58]. Since sea ports play such a central role in the supply chain these are discussed in more detail.

Not a single sea port container terminal is the same. There are differences in size, function and geometric layout, but the subsystems of a terminal are all the same. A terminal is built up from a ship operation area, a yard and an area for hinterland operations. The ship operation area is also called the berthing area. In this area vessels dock at a berth and are (off)loaded by quay cranes. In the yard the containers are stored that are waiting for pick up by a vessel or hinterland transportation mode. Containers that need cooling or store hazardous materials are stacked in special areas. Also empty containers have their own yards. In the hinterland operations area containers are loaded on or offloaded from their hinterland transportation mode: train, barge or truck. So for example an import container follows this chain of operations:[18, 46, 58, 85]

1. The container vessel arrives at the port.
2. A berth is assigned to the vessel.
3. The container vessels docks at its berth.
4. The container is offloaded from the vessel.
5. The container is distributed to a storage block in the yard.
6. The container is stored in its storage location, given by row, bay and tier. This location is assigned in real time upon arrival.
7. The container is retrieved from the yard when his next transportation mode arrives.
8. The container is loaded on the next mode of transportation and transported to the next destination.

Since hundreds of containers are offloaded from or loaded on a vessel[92], a huge number of concurrent operations with the different transportation modes and handling equipment needs to be scheduled and this task is extremely complex. Furthermore the conditions of the terminal change every second and future events and their timing can be predicted to a certain limit, so control task need to be solved in real time[18]. The operations that are being carried out are strongly interrelated[13]. So the arrival of a vessel influences every following step in the chain of operations and thus influences the planning at the terminal.

A sea port container terminal can be described as a complex system with incomplete information about events in the future and with interactions that are highly dynamic in nature between handling, transportation and storage units[13, 18]. In this system a lot of decision problems exist that are related to logistics planning and control issues[18]. These problems become unmanageable without the help of proper methodological tools because of the complexity in the system[13]. These problems are divided into three categories: terminal design, operative planning and real time control[18]. Since the terminals are already built, only the relevant problems of the terminal design category are discussed and focus is placed on operative planning and real time control.

Terminal design

During the initial planning stage of a terminal the design problems need to be solved by the facility planners. Design problems can be for example the terminal layout, equipment selection, berthing capacity, multi-modal interfaces and the IT systems and control software[18]. The problems regarding IT systems and control software are highlighted.

IT systems and control software Logistics control is an incredible complex task in a container terminal, which requires decisions made in real time with regard to matching the handling task with the corresponding unit of equipment while providing information about every single container. Therefore using sophisticated tools for optimization and also using different modes of software and IT support are of utmost importance[18].

Operative planning

Operative planning consists of guidelines and basic planning procedures for performing the various logistic processes at the terminal. Decentralized planning is the only mode that makes it possible to govern the logistics control of an automated container terminal. Therefore the logistics control system is divided into various subsystems related to the type of resource needed for the control. The planning of these resources is done on a short-term horizon of several days because specific problems arise in scheduling and planning[18]. The relevant different subsystems are discussed.

Berth allocation Before a vessel arrives a berth needs to be assigned to the vessel. This is constrained by the time of arrival, the time needed, availability of berths, availability of handling equipment, the requirements for the cranes and the requirements of other vessels that are at the quay or arrive shortly after[18].

Workforce scheduling An important resource in container terminals is the workforce. The workforce operates the equipment and their rosters and schedules need to be generated in advance[18]. The workforce at a container terminal is usually scheduled in four shifts of six hours[13]. This planning is done on two levels, monthly and daily[47]. The monthly schedule makes sure that the required personnel is available every working day conforming to arrangements, contract obligations and regulations. Since the demand for the workforce is uncertain, two types of workers exist in the daily planning. Fixed workers that are assigned to a specific shift and flexible workers for who the shift is determined during the daily scheduling when more information about the demand is available. The different scheduling levels are characterized by process complexity, the temporal fragmentation and an increasing information uncertainty when the time horizon increases[13].

Real time control

Container terminals are considered as a highly stochastic and dynamic logistics system, that only allow a look-ahead horizon of a maximum of 10 minutes for pre-planning of detailed transportation and handling activities. So real time control of the logistic activities is of extreme importance. The real-time control is activated by an event or condition and it is necessary that the related decision problem is solved very quick, preferably within seconds. The decisions can be the assignment of a transportation order to a vehicle, the scheduling and routing of the hinterland transport, transportation between handling areas and the yard and many more[18].

Now the operational problems in the supply chain are discussed, disturbances in the planning are identified as a problem. In the following the consequences of these problems and possible solutions are presented.

Since the supply chain is a network of interdependent activities, the disturbances in the planning of a sea port terminal are not only limited to the sea port terminal. All the stakeholders involved in the supply chain experience negative effects due to these disturbances at the sea port[58, 79]. The biggest source for disturbances is the uncertainty about the actual arrival time of a vessel at the port[15, 58, 81]. If a vessel arrives late, the unloading is delayed and consequently the assignment to a hinterland transportation mode is delayed. As a consequence the delivery may be delayed at the final destination[58, 81]. The uncertainty about the arrival time of a vessel also leads to a huge uncertainty about the demand over a specific time period at the port[58]. This uncertainty for demand at the port influences the demand for resources like the workforce as is identified in the workforce subsystem section. So the task of a workforce planner is much harder due to this uncertainty and also the effectiveness of the planning diminishes because of this uncertainty[13]. Another problem, as identified in the operative planning section, is the berth allocation of a vessel. Due to the uncertainty about the arrival, allocating berth places to vessel is extremely complex[58]. As noted the activities in the network are dependent on each other, so since the actual time a container is offloaded is characterized by uncertainty, the planning of the next transportation stage can not be optimized, so booking extra capacity on barges or trains, because of the limited availability of scheduled departures, and ad-hoc usage of trucks is necessary which lead to an increase in the cost, longer transit times and can negatively impact the security level[58, 68, 92]. Another conservative approach is using slack times in the planning, where some wait time at the terminal is planned, to avoid late deliveries[81, 91, 92].

The uncertainty about the arrival time stems from the possibility of unforeseen events and poor forecasting[13], since the ETA is guessed by the crew of a vessel[15]. So although contractual obligations exist to communicate the ETA 24 hours before arriving at the port, often the ETA is revised within this time period[13]. These revision are also mandatory to notify partners in the supply chain about exceptions or deviations on time[70]. When we take the problems, that are the consequence of the uncertainty about the arrival time, and the poor forecasting method into consideration, it becomes clear that forecasting the ETA needs to be improved and also can be improved upon. With a more reliable prediction of the ETA the resources can be allocated more efficiently and the hinterland transportation planning can also be improved upon[13, 58, 92]. This also lowers the cost in the supply chain[91, 92].

3

Literature review

In this chapter the literature review is presented. First the benefits of and problems with information exchange in container transport are discussed. An increase in information has beneficial effects, AIS data is identified as a useful source of information. The characteristics of AIS are discussed and also possible problems with AIS that needs to be accounted for. An overview of AIS related research is provided with a short description of the research. From this overview is concluded that ETA forecasting is hardly researched upon. The overview also shows that trajectories of vessels can be derived from historical data and used to predict the route of incoming vessels. Last the problem is scoped and the research questions are presented.

3.1. Information exchange

As discussed in Section 2.2 every following step in the supply chain is dependent on the time of arrival of the container vessel. So in order for a hinterland network to be efficient it is key to receive accurate information about the arrival time of the goods on a container vessel[81]. If a forecast is incorrect the negative effects impact the entire supply chain and the effects aggravate at every link in the supply chain, this is called the bullwhip effect[39]. However the effects can be mitigated if information is exchanged throughout the supply chain and not kept in functional silos[36, 90]. But since stakeholders in the supply chains are reluctant to share information in order to keep their competitive advantage[36], publicly available is relied upon.

One stakeholder that can play a vital role are seaports. Seaports are able to connect global networks to increase the visibility with the use of information technologies and thus they increase the efficiency, security and environmental friendliness of global supply chains. Also with the use of information systems the reliability in the supply chain increases[90]. Furthermore when seaports have access to more (accurate) information, such as an Estimated Time of Arrival(ETA), stakeholders at the port are able to improve on their planning activities and as a result lower the cost in the supply chain because for example less ad-hoc truck transport needs to be arranged[91, 92].

As in many other industries, Information and Communication Technology(ICT) is considered in freight transport as a major enabling technology to improve the planning[70] with the use of descriptive, diagnostic, predictive and prescriptive analytics[36]. Reliable data capture, storage, processing and communication, via electronic means, are vital to use these technologies. Possible improvements in supply chains are dependent on this collection, analysis and communication of information[36]. The advancements in ICT, both hardware and software, have made ICT technologies available on the market at reasonable prices. And although a lot of ICT technologies are being used for certain applications, they can be used in new ways. For example machine learning can be used to predict the ETA[12, 13, 58]. In order to use ICT technologies it is important that a standard is determined. When a standard for data exchange and data content or messages is established, the data becomes interoperable between the different collaborators in the supply chain[70].

The International Maritime Organization (IMO) have imposed a standard for container vessels, this standard is called Automatic Identification Standard (AIS)[74]. This standard is discussed in Section 3.2. Just like containerization revolutionized maritime shipping, automatic tracking via AIS now enables major advancements in logistics like efficiency gains, increased security and improved customer service[36]. The AIS sends vessel navigation information that could be used to identify navigation patterns[74]. But the AIS could also function as a sort of Track & Trace system for vessels[36, 72] and this system can be used for arrival or delay notifications regarding particular vessels[36, 70]. These delays need to be communicated to other partners in the supply chain, in order for them to make changes in due time[70]. Preferably partners should be able to track shipments, monitor performance and react in real time to optimize the freight trip[36].

So every partner in a supply chain should be provided with as much information as possible to improve the performance of a supply chain. However to keep their competitive advantage, stakeholders are reluctant to share information in the supply chain. Therefore publicly available is used in the research. One of the information streams that can be generated with the use of ICT and is helpful in the supply chain, is information regarding the ETA of a container vessel in a certain port. These ports are also vital in connecting the global supply chains, they are the link between the maritime transport and the hinterland and can provide the ETA to planners to improve on the transportation in the hinterland. This ETA can be computed with machine learning, preferably in real-time, using data that is send using the AIS infrastructure[58]. Therefore AIS is discussed in the next section.

3.2. AIS data

The Automatic Information System, also known as AIS, has been introduced by the International Maritime Organization(IMO) in 2000. Every vessel that conducts international voyages with a tonnage of 300 and upwards or vessels with a tonnage above 500 are obliged to have AIS installed since 2005. Thus becoming the standard in real-time vessel to vessel and vessel to shore communications[1, 26, 58, 60, 72, 83, 87]. Furthermore in the European Union, fishing vessels with a length above 15 meters are also obliged to have AIS installed[53]. The aim of AIS is to increase safety and navigation efficiency, tracking, and improved situational awareness and assessment[20]. AIS exchanges information electronically and creates the possibility of tracking and automatic identification, from shore stations and also from other vessels[30, 36, 60, 72]. AIS is self-reporting and its primary use was to avoid collisions[20, 56]. AIS provides updates on the dynamic information regarding the vessel on different time intervals, depending on the speed of the vessel[1, 17, 26, 34, 35, 60, 72, 74, 87]. An overview can be found in Table 3.1. Static and voyage information is updated every 6 minutes[17, 26, 74]. We discuss the AIS messages in more detail in Subsection 3.2.1. The information is send over a special Very High Frequency(VHF) radio frequency to the Vessel Traffic Services(VTS)[20, 74]. AIS messages consist of state vectors and identity information[83]. The dynamic information of a vessel is obtained with technical instruments, for example the location is determined using an embedded GPS unit with a accuracy of 10 meters[58, 60, 74]. Static and voyage related information needs to be inputed manually[58]. AIS is one of the main information sources in maritime surveillance[17].

Table 3.1: Broadcast intervals of dynamic AIS data, derived from IMO[26]

Speed	Changing course	Broadcast interval
At anchor	No	3 minutes
0 - 14 knots	No	12 seconds
0 - 14 knots	Yes	4 seconds
14 - 23 knots	No	6 seconds
14 - 23 knots	Yes	2 seconds
23+ knots	No	3 seconds
23+ knots	Yes	2 seconds

So all the information from the AIS is received by the VTS, giving them an enormous amount of information that has a potential great importance and value regarding container vessels[74]. Identification, tracking and vessel monitoring activities of the VTS are enhanced by the integration of AIS into their systems[1, 20, 74]. However the information obtained by the VTS is not fully exploited[74]. The received spatiotemporal data is stored in databases so it can be used in research[35, 56, 74, 89]. These databases are huge and contain data regarding for example about the position, heading, speed over ground and rate of turn[23]. Since AIS is a worldwide standard, it provides a coherent source of information that is near real-time for maritime traffic analysis with global coverage[24, 41, 83], at this moment latency is about 1 minute[54].

To gain insight about the content of AIS messages and how it can be used in the research the content of AIS messages is discussed in the next section.

3.2.1. AIS messages

AIS messages are send via a VHF frequency to other vessels and inland stations[20, 26, 60, 87, 89] and also to satellites[24, 41, 83]. These messages are sent on different intervals as identified in Table 3.1. Because AIS uses VHF, AIS is able to detect other vessels using AIS while radar in the same situation is not able to detect these vessels. This is possible when a vessel is around a bend, behind a hill or when weather conditions, such as fog or rain, restrict the visibility[20]. The AIS system onboard a vessel is a transceiver system[60], making it possible to both receive and sent AIS messages. These transceivers broadcast messages autonomously and continuously[40, 60] and these messages are sent in every direction[41]. These messages can be received by a transceiver on other vessels or receivers that are fairly cheap[20]. Also noted in the previous section, the dynamic information is compiled by the vessel's instruments that are connected to the AIS transceiver[1, 58, 60, 74]

Messages sent with the AIS system consist of three types of information: Static, dynamic and voyage related. A list of the information included in AIS messages is compiled[1, 2, 4–6, 6, 8–11, 17, 20, 23, 24, 26, 30–32, 34, 35, 37, 38, 40–42, 52, 57, 58, 61, 62, 65, 66, 72, 74, 75, 83, 84, 87, 89]:

- Static information:
    1. IMO and Maritime Mobile Service Identity (MMSI) number
    2. Call sign and name
    3. Type of vessel (passenger, tanker, etc.)
    4. Length and beam
    5. Location of position fixing antenna such as GPS/DGPS (aft of bow, port or starboard of C/L)
- Dynamic information
    1. vessel's position with accuracy indication (for better or worse than 10 m) and integrity status
    2. Time in UTC (coordinated universal time)
    3. Course over ground (COG)
    4. Speed over ground (SOG)
    5. Heading
    6. Navigational status (e.g., not under command, constrained by draught, etc.)
    7. Rate of turn (where available)
    8. Angle of heel (optional)
    9. Pitch and roll (optional)

- Voyage related information
    1. vessel's draught
    2. Type of cargo
    3. Destination and estimated time of arrival (at master discretion)
    4. Route Plan-waypoints (optional)
    5. Number of persons on board (on request)
- Short safety messaging
    1. Short text messages with important navigational safety related information are shown in an extra window.

3.2.2. Problems with AIS

Although AIS is a reliable information source for maritime traffic, it also comes with some problems[12, 20]. AIS is known to sent data inconsistencies and anomalies about the maneuvers of a vessel[83]. Furthermore spoofing, where false information is intentionally sent, is a problem[31, 83]. But because the research covers shipping lines, the possibility that data is spoofed is very small and logistics providers claim that it is not a problem[12]. Therefore spoofing is not taken into consideration in this research. In the early days of AIS, losses of position reports were a problem[60]. However after these researches were conducted a lot of improvements to the network have been made regarding the coverage and findings from the literature review indicate that since these improvements no coverage problems have occurred.

Next to these problems, also the content of the AIS messages can contain manny errors. This has been researched upon by Bailey[4], Harati-Mokhtari et al.[20] and Norris[52]. The possible errors in AIS messages are discussed in the next section.

Static information

Static information is entered when the system is installed[4, 20, 52]. So it might be surprising that errors in this information occur. However installation is done under time pressure with bad training[4, 52] so data is not entered properly.

MMSI number Every vessel has an unique number for AIS identification number, the Maritime Mobile Service Identity (MMSI) number. These numbers are entered into the AIS on installation. However some vessels still transmit the default MMSI number 1193046[73]. Another possible error in the MMSI number is that the vessel transmits the number 0, 1 or 999999999. It is also possible that a vessel transmits a number that is less than the mandatory 9 digits[20].

vessel's name and call sign Problems with the vessel's name were encountered in the research of Harati-Mokhtari et al.[20]. In 6% no name or call sign was transmitted by a vessel. Another problem is the use of abbreviations in the vessel's name, causing possible confusion about the name of the vessel. These abbreviations are caused by the limit of 20 characters available in this field.

Vessel Type The vessel type is selected from a predefined list upon installation. However in some cases the vessel type field is left blank or are simply called "vessel". Another problem are vague or misleading vessel types. Vessel types for example are kept general, like cargo vessel when dealing with a tanker. This problem is caused by a limited number of predefined categories on the one hand and on the other hand it is not feasible to include every potential vessel type since some very distinctive vessels travel our waters[20].

Length and beam Errors in length and beam consist of not displaying the information, displaying incorrect information or incorrect correlation between length and beam[20].

Dynamic information

Position Harati-Mokhtari et al.[20] have not looked at the practical accuracy of AIS. However they did notice that in a few cases positions were transmitted that were not possible. Like a latitude above 90° or longitude above 180° or the position 0° N/S, 0° E/W.

Vessel navigation status The navigation status needs to be manually updated by the officer of the watch or the navigation officer. In the system in 2007 crosschecking was not incorporated in AIS, causing vessel to display the incorrect status because updates were not entered[20].

Voyage related information

Voyage related information needs to be entered and updated manually for every voyage. This is not always done properly[58] or not at all[52].

Draught In the researches of Bailey[4] and Harati-Mokhtari et al.[20] it was found that vessels did not report a draught or a draught of 0 meter. Also a few cases were found where the draught was off by several meters, which is a big difference. And might be the difference between grounding or not grounding a vessel.

Destination and ETA About half of the vessels in the research of Harati-Mokhtari et al.[20] displayed wrong information in this field. Possible errors in the destination field are[4, 12, 20]:

- A number instead of a destination
- A country instead of a port
- Abbreviated names
- Showing not available, not defined or NULL
- Fake input, e.g. to hell
- No input
- Previous port as destination

Furthermore problems with the ETA are not updated ETA's, ETA's in the past or ETA's in a very distant future[20].

So possible problems in the AIS data are identified. In order to guarantee the quality of our data it is vital that the quality of the data is checked and any possible problems are solved in pre-processing[12].

3.2.3. Uses of AIS

Although AIS was first introduced for radar augmentation and VTS, it is also used to gather information about maritime traffic around the globe[1]. In maritime traffic research AIS is chosen as a data source[87, 89] for a few reasons: (1) The good reliability and availability, (2) AIS data is neutral and carries hardly any subjective distortion and (3) AIS data is well stored in databases[87]. Research on AIS data is conducted to discover new values from the enormous amount of data with statistical models, data mining[89] or data fusion algorithms[31]. For the research it is important that the data has an efficient representation and consistent knowledge of the behavior of a vessel[83]. With the introduction of AIS a lot of information about trajectories of maritime trajectories have become available[23, 40]. This data can be analyzed so normal behavior patterns can be identified. However due to the huge amount of information available it is an incredible effort to do this manually, therefore efficient and robust automatic data processing should be used to process the information and possibly generate input for manual investigation or operational decisions[35, 57]. The size of the datasets is also an advantage, since the statistical confidence of findings increase with the size of datasets[34]. Table A.1 in Appendix A shows an overview of research that is related to AIS. This list is not complete, but provides a nice overview in different subjects of AIS research. A lot of research into AIS is conducted, especially in the period 2010-2013.

The researches from the overview are categorized based on their subject in Table 3.2. The researches are not discussed on a very detailed level but a categorization of the researches and their methods is provided. After this overview a more detailed discussion of relevant researches is presented. As can be seen in the table the researches are categorized in five categories: Generic uses & performance, motion patterns, anomaly detection, prediction and collision avoidance. The first category Generic uses & performance comprises research into subjects such as the basic principles of AIS[8, 84], errors in AIS[4, 5, 16, 20, 31, 52] as discussed in Subsection 3.2.2, AIS coverage[24] and database management for AIS data[10]. The second category consists of researches that cluster the motions of different vessels to find standard motion patterns of vessels. Anomaly detection is related to researches that try to identify behavior of vessels that is not normal, for example deviating from the standard motion patterns. Researches in the category prediction focus on ETA prediction over a multi-day time span[58], movement prediction over a short time span[6, 60, 61, 89] or route prediction[12, 35, 56]. The last category contains researches that try to predict collision and use these predictions to avoid them[23, 72] and of research that uses AIS data to evaluate collisions that already occurred.

Table 3.2: AIS research by category

Generic uses & Performance	**Motion patterns**	**Anomaly Detection**	**Prediction**	**Collision avoidance**
Chang 2004[8]	Bomberger et al. 2006[6]	Bomberger et al. 2006[6]	Bomberger et al. 2006[6]	Hornauer & Hahn 2013[23]
Graveson 2004 [16]	Rhodes et al. 2007[61]	Tun et al 2007[75]	Rhodes et al. 2007[61]	Talavera et al. 2013[72]
Bailey 2005[4]	Tun et al. 2007[75]	Ristic et al. 2008[62]	Redoutey et al. 2008[60]	Wang et al. 2013[87]
Harati-Mokhtari et al. 2007[20]	Ristic et al. 2008[62]	Laxhammar et al. 2009[38]	Lane et al. 2010[35]	
Norris 2007[52]	Aarsæther & Moan 2009[1]	Lane et al. 2010[35]	Pallotta et al. 2013[57]	
Baldauf 2008[5]	de Boer 2010[9]	Laxhammar 2011[37]	Wijaya & Nakamura 2013[89]	
Høye et al. 2008[24]	Demšar & Virrantaus 2010[11]	Kowalska & Peel 2012[32]	Dobrkovic et al. 2015[12]	
Redoutey et al. 2008[60]	Lane et al. 2010[35]	Vespe et al 2012[82]	Parolas 2016[58]	
Vespe et al. 2008[84]	Lampe et al. 2010[34]	Vespe et al. 2012[83]		
Guerriero et al 2010[17]	Lei et al. 2011[40]	Pallotta et al. 2013[57]		
Tsou 2010[74]	Sampath 2012[65]	Scholte 2013[66]		
Katsilieris et al. 2013[31]	Vespe et al. 2012[82]			
Liu & Chen 2013[41]	Vespe et al. 2012[83]			
Ma et al 2013[43]	Pallotta et al. 2013[57]			
Loptiën & Axell 2014[42]	Talavera et al. 2013[72]			
de Vreede 2016[10]				

The researches in the categories motion patterns and prediction might be useful for the research. Researches of interest are researches that predict the ETA over a multi-day timespan and researches that predict the route of vessels. These researches from Table 3.2 are discussed below.

Lane et al. 2010[35] This research expands on the research of Nevell[51]. Nevell introduced networks and bayesian theory to predict the route of a vessel. Lane et al. expand on this research with port visitation patterns. These patterns are modeled with Hidden Markov Models. Since ports are visited in a particular order they are able to do this. With this method the possible routes that a vessel takes are constructed. Therefore Hidden Markov Models are incorporated in the research. Next to Lane et al. Hidden Markov Models are also used by Guerriero et al. and Tun et al. Guerriero uses the Hidden Markov Model to detect anomalies in AIS messages[17]. Hidden Markov Models are used by Tun et al. to model the movements of a vessel in a port area, so it's more or less route prediction but their time horizon is very short[75]. But is has been shown that Hidden Markov Models can be used in combination with AIS data.

Pallotta et al. 2013[57] Pallotta et al. construct vessel routes with the use of waypoints with the Traffic Route Extraction and Anomaly Detection(TREAD) method. With these routes and historical information they are able to predict the possibility that a vessel is following a certain route and they are also able to predict possible routes of the vessel. Although they only predict the next destination of a vessel expanding this method might be possible to identify every possible route that has the Port of Rotterdam as a final destination.

Dobrkovic et al. 2015 [12] Dobrkovic et al. reviewed several papers that try to predict the routes of vessels. They identified 4 areas that need to be improved upon: data quality, data volume and distributed data mining, discovery and inclusion of behavioral patterns & fusion of weather data. They provide recommendations how to improve on this area. Their recommendations are incorporated in the research.

Parolas 2016[58] The research of Parolas is the only research that has the same focus as this research. It aims to predict the Estimated Time of Arrival(ETA) of container vessels. The research shows that on the Shanghai - Rotterdam route the ETA can be predicted for vessels directly traveling to Rotterdam over a time horizon of five days. It is more accurate than the ETA communicated via AIS and is predicted using Neural Networks and Support Vector Machines. Also Parolas incorporated weather prediction in his algorithm, however his research shows that weather predictions have no influence on the ETA for this specific route.

3.3. Other relevant research

Fancello et al. 2011[13] Another research that predicts the ETA of container vessels is the research of Fancello et al. They show, just like Parolas[58], that with a neural network a more accurate prediction can be made regarding the ETA of a container vessel. However a lot about their research is unclear. They do not state where their dataset comes from. They also fail to mention over which time horizon they predict the ETA and for which shipping route they predict the ETA. So although they claim that they can provide a more accurate prediction of ETA for every vessel independent on the port, their claim can not be checked. However the research provides useful insights.

3.4. Problem scoping

The review shows that only two researches have been able to forecast the ETA of container vessels. Fancello et al.[13] were able to predict the ETA with a neural network however it is not clear over which time horizon and to which type of shipping lines but maybe more importantly it is not clear where their data comes from. So although a proof of concept is provided, cheking if their method provides the results they claim is impossible. They do state however that the algorithm is generic and can be applied to all ports.

Parolas[58] was able to predict the ETA of vessels directly shipping to Rotterdam over a 5 day time-horizon. However not all vessels travel directly to Rotterdam. It is common practice in container shipping to visit multiple ports on one route. Since this problem has not yet be researched upon, the research is scoped to this problem.

Since vessels visiting multiple ports are the focus point of the research, information about their routes needs to be generated. Because AIS only contains information about the next port of destination, information about the routes needs to be generated. However in the literature Hidden Markov Models and Bayesian theorem in combination with network modeling, e.g. Nevell and Lane et al. [35, 51] are used to predict the routes of a container vessel. Pallotta et al.[57] propose the TREAD method and use this method to predict routes for vessels, but this methods only predicts the next destination. From these methods valuable insights are taken to construct a route identification method. Route identification is combined with the aforementioned ETA forecasting techniques to predict the ETA of vessels visiting multiple ports on their routes before arriving in Rotterdam.

3.5. Research questions

In Chapter 1 the research question is introduced:

How to improve the AIS-based ETA predictions of vessel en route to a port by leveraging route identification?

The main research question is split into multiple sub-questions based on the sub-objectives. To do the ETA prediction and route identification, data that meets certain quality standards are needed. With the first research question possible issues with the data quality are identified and used to construct a framework to pre-process the AIS data, to meet the data quality standards, so it can be used for ETA prediction and route identification. These steps are identified to answer the second research question. In the research it is hypothesized that each route has characteristics that influence the travel time of a container vessel and thus the arrival time. Therefore all the possible routes for a vessel are identified. How to identify these routes is the answer to the third research question. Since it is hypothesized that every route has different travel times, incorporating the routes into the ETA predictions is needed. It is showed how the routes are combined with ETA prediction to answer the fourth research question. Last the added value of an ETA prediction tool is discussed to answer the last research question. To answer this question a stakeholder analysis is performed and the added value for the stakeholders is discussed. Also the possibilities for a stakeholder to influence the data quality of AIS messages is discussed.

1. *What are possible issues with the data quality of AIS messages?*
2. *How can AIS messages be pre-processed to improve the data quality so it can be used for route identification and ETA prediction?*
3. *How can a set of possible routes of a container vessel be identified using pre-processed AIS data?*
4. *How can the ETA of a vessel be predicted with the use of pre-processed AIS data and route identification?*
5. *What is the added value of the proposed algorithms and framework for stakeholders in the supply chain and how can stakeholders influence the data quality of AIS messages?*

4

Methodology

In this chapter the methodologies used by Tsou[74], Gomez et al.[15] and Fancello et al.[13] are introduced. The steps are discussed and these methods are compared. The comparison is used to design a methodology that comprises ETA prediction and route identification.

4.1. Data mining and AIS data

In this section the methodology as introduced by Tsou[74] is shortly discussed. The methodology focusses on data mining and is based on the work of Roiger and Getaz, 2002[64] and the work of Han and Kamber, 2011[19]. The methodology consists of seven steps as discussed in Table 4.1.

Table 4.1: Methodology of Tsou[74] based on Roiger and Getaz, 2002[64] & Han and Kamber, 2011[19]

Step	Description
Set the target	To generate an understanding of the domain where data is going to be minded, it is necessary to clearly describe the objectives and compile a list of assumptions and the results that are anticipated.
Establish target dataset	Determine the dataset that you want to use for analysis, such as an AIS dataset.
Data pre-processing	Use approaches that are effective and ready to use for the processing of noisy data. Also decide how to handle data loss.
Data cleaning and transformation	In this step attributes and information are deleted or added. Also determine the methods to standardize, convert and modify the data. Convert the dataset to a format suitable for data mining and store the dataset.
Data mining	Use data mining algorithms that are appropriate to process the data.
Explanation and evaluation	Use the results to gather useful and interesting information. If no information is available repeat the process adding other new attributes and samples.
Action	If the information found ins step 6 is perceived as useful, use the information to solve the problem.

4.2. Forecasting in Container Terminal Operations

Gomez et al.[15] introduced a methodology that uses multiple steps for forecasting. These steps are discussed in Table 4.2.

Table 4.2: Methodology of Gomez et al.[15]

Step	Description
Parameter selection	Select the parameter that needs to be forecasted.
Identify influencing factors	The parameter that needs to be forecasted can be influenced by a lot of factors such as speed and position. In this step these factors are identified, the identification is done with the help of experts or from literature review.
Identify climatic and operational drivers that produce variations in influencing factors	This step is used to identify phenomena that can influence the influencing factors, such as the weather that may be able to influence the speed of a vessel.
Collect historical data on climatic and operational drivers	Since these drivers have an effect on the influencing factors, historic data is needed to train the machine learning technique.
Data preprocessing	Transform the data in a format the is readable for the machine learning technique. Tasks include: verifying, editing and editing the dataset to remove errors, remove outliers or replace missing values. For example variables may be normalized.
Configure machine learning technique	Each technique has parameters that can be set. The performance of the technique is influenced by these parameters. In this step the parameters are set.
Train	Feed a sample of the dataset with an entry for the parameter to the machine learning technique. This dataset is used to train the machine learning technique. This produces a model that can be used for prediction.
Evaluate the model	To evaluate the performance of the model error measures are used, error measures can be individual forecast error, mean error, mean square error, mean absolute error, mean absolute percent error, or mean absolute scaled error[25]. Evaluation can be done on a different validation sample of the dataset.
Retrain the model	When the performance of the model is not satisfactory the machine learning technique needs to be retrained. It is also possible to reset the configuration of the technique.
Save the configuration	When the results are satisfactory the model is saved and used to predict the parameters

4.3. Forecasting the ETA of a vessel

In Table 4.3 the methodology used in the research of Fancello et al.[13] is presented. This methodology is based on the Phd thesis of Pisano[59] who is part of the research team. They used a neural network in their research so this methodology concerns neural networks. However they do not use AIS data as input for the model.

Table 4.3: Methodology of Fancello et al.[13] based on the work of Pisano 2008[59]

Step	Description
Choice of predictive approach	Choose a predictive approach for the neural network
Choice of paradigm	Choose a paradigm for the neural network.
Choice of input variables	Choose the input variables using a priori knowledge.
Data pre-processing[1]	Add extra variables to the data base using a priori knowledge. Check correlations between variables and eliminate strongly correlated variables.
Variable normalization	Normalize the variables for use with the algorithm.
Choice of network architecture	Divide the dataset in a training and validation set. Use the validation set after training to evaluate the predictive ability.
Choice of number of hidden layers and nodes	Set the number of hidden layers and nodes using trail and error.
Choice of learning algorithm and related parameters	Choose the learning algorithm and set the parameters of the algorithm.
Interpretation of results	Interpret the results of the algorithm.

Fancello et al. state in their research that the predictive capabilities with this methodology were not as expected. They believe this is because of a shortcoming in the choice of input variables. Therefore they added a data pre-processing step. They add an extra variable using a priori knowledge and after that they analyze the correlations between the variables. Strongly correlated variables are eliminated since these do not provide additional information. The remaining variables are analyzed with multivariate statistical techniques. The remaining variables were reorganized depending on their significance. The results of the significance test were used to determine four sets of alternative inputs to the neural network[13].

[1] This step was added later to the methodology

4.4. Comparison and methodology development

In this section the three methodologies that are discussed in this chapter are compared. The comparison is used to develop a methodology. The comparison is shown in Table 4.4.

Table 4.4: Comparison of the different methodologies

Tsou	Gomez et al.	Fancello et al.
Set the target	Parameter selection	
		Choice of predictive approach
		Choice of paradigm
	Identify influencing factors	Choice of input variable
	Identify climatic and operational drivers	
Establish target dataset	Collect historic data on climatic and operational drivers	
Data pre-processing	Data pre-processing	Data pre-processing
Data cleaning and transformation		Variable normalization
Data mining		
Explanation and evaluation		
Action	Configure machine learning technique	Choice of network architecture
		Choice of number of hidden layers & nodes
		Choice of learning algorithm and related parameters
	Train the model	
	Evaluate the model	Interpretation of results
	Retrain model	
	Save the configuration	

As discussed in Subsection 3.2.3 a possible set of routes is identified for container vessels. So next to the steps identified in Table 4.4 also steps are needed to construct this model. Some steps from Fancello et al. are omitted because they focus specifically on neural networks. These steps might become important when neural networks are used, but the methodology is kept as generic as possible. Also the data mining and ”explanation and evaluation” step of Tsou are not incorporated in the methodology since these are related to data mining and the data is not mined. The other steps identified in the methodologies are all incorporated in the methodology. An overview of the methodology is presented in Table 4.5 and every step is discussed in the following.

Set the target In this step the target value is set. This value is related to the problem statement of the research.

Identify input variables In this step the input variables are identified that may have an influence on the target value. These variables are identified using a literature review.

Collect dataset A dataset is collected that has information regarding the input variables and the target value.

Data exploration The dataset is explored to gather knowledge about the dataset and to identify possible problems that might occur as discussed in Subsection 3.2.2. For example the positional data is checked to be within the possible limits.

Data pre-processing The data is pre-processed before using it. Extra variables are added that can be calculated from the information, such as the size of the vessel. Furthermore the data is standardized for processing if necessary.

Route identification Insight from the Hidden Markov Model[35] and TREAD method[57] are used to design a route identification algorithm.

Choose machine learning technique A machine learning technique is selected to be used for predictions on the dataset. This choice is be based on a discussion of the different available techniques and their suitability to the problem at hand. Since the next step depends on the chosen technique.

Configure machine learning technique The parameters of the selected machine learning technique are configured to obtain the results.

Train models Training sets are constructed for every routes. In this step the training sets are fed to the machine learning technique to develop a model for every route that can be used for prediction.

Evaluate models Using the test set consisting AIS messages of all routes and error measures, the performance of the models are evaluated.

Retrain model When the performance is not satisfactory the machine learning technique is reconfigured and retrained for a new model. This step may also be repeated at regular time intervals to incorporate new data.

Save model When the model performs as wanted the model is saved and used for prediction.

Table 4.5: Methodology for this research

Step	Description
Set the target	Decide target to predict.
Identify input variables	Identify variables that influence the target.
Collect dataset	Gather data or dataset to be used.
Data exploration	Explore the dataset.
Data pre-processing	Process the data so it can be used for prediction.
Route identification	Design a route identification algorithm.
Choose machine learning technique	Choose what technique to use.
Configure Machine learning technique	Set the parameters of the machine learning technique.
Train model	Feed the training set to the machine learning technique.
Evaluate model	Evaluate the performance of the model.
Retrain model	Retrain if necessary.
Save model	Save the model.

In the methodology the data exploration and data pre-processing steps are used to answer the first and second research question. The route identification step is related to the third research question. In this step is showed how the possible set of routes of a container vessel can be identified. The following steps are used to show what machine learning techniques are feasible for the research and how to incorporate the possible set of routes for a vessel into the algorithm. The incorporation of the possible set of routes into algorithm answers the fourth research question.

5

Improving data quality of AIS messages

In this chapter the first two sub research questions are answered: "What are possible issues with the data quality of AIS messages?" and "How can AIS messages be pre-processed to improve the data quality so it can be used for route identification and ETA prediction?". The requirements for the dataset are set in this chapter and possible data quality issues are identified. Furthermore the framework, as shown in Figure 5.1, to improve the data quality is introduced. The steps in this framework are discussed in the chapter.

5.1. Target setting and input variables

In order to make a prediction, a target must be set first. This is target setting. The target is set based on the research question of this thesis. The research question is:

How to improve the AIS-based ETA predictions of vessel en route to a port by leveraging route identification?

The goal of this research is to improve the accuracy of ETA prediction for vessels that are en route to a port using AIS data with the Port of Rotterdam as a case study. So the target to predict will be the ETA of vessels at the Port of Rotterdam. This target is predicted with the use of AIS data so in the data collection step AIS data is collected.

Now the target has been set to predict, the input variables are identifed that might have an influence on the ETA of a vessel. Since the research of Fancello et al.[13] and Parolas[58] also predict the ETA of a container vessel in their research their input variables are evaluated to see which might be needed in the research. Furthermore some variables are added that have an influence on the ETA of a container vessel.

5.1.1. Input variables of Fancello

Fancello et al. has identified 9 variables that could be used as input for ETA prediction: 'ship name', 'ship length', 'transit time', 'number of dockers required for unloading', 'number of dockers required for loading', 'ETA month', 'ETA day of the week', 'ETA hour' and 'ship's port of departure'. They performed a correlation analysis on these variables to eliminate redundant variables. The remaining variables were combined into four sets of input which are shown in Table 5.1[13].
Their results show that the second set of variables results in the best predictions[13]. So as possible inputs for the models can be used: The name of the vessel, the port of departure of a vessel, the number of dockers required for loading.

Table 5.1: Input sets in the research of Fancello et al.[13]

Variable	Set 1	Set 2	Set 3	Set 4
Name of ship	X	X	X	X
Departure port	X	X	X	X
Crew loading	X	X	X	X
Crew unloading	X		X	X
ETA month	X			X
ETA day of week	X			
ETA hour	X			

5.1.2. Input variables of Parolas

Table 5.2 presents an overview of the input variables that were used by Parolas for predicting the ETA of container vessels[58]:

Table 5.2: Data used as input variables for predicting the ETA of vessels[58]

AIS Data	Weather Data
Latitude (degrees)	Current U-Component (m/s)
Longitude (degrees)	Current V-Component (m/s)
Distance to be covered (km)	Wind U component (m/s)
Current Speed of the vessel (km/h) COG	Wind V component (m/s)
Change in speed over the last 3 hours (km/h)	Peak wave period (s)
Average speed based on last 12 hours (km/h)	Peak wave direction (degrees).
Time used for calculating the average speed (hours)	Significant wave height (m)
Length of the ship (meters)	
Breadth of the ship (meters)	
ETA of the ship's agent (number of days)	

One of the conclusions of the research of Parolas was that weather does not influence the ETA prediction[58]. So weather data is not used in these research. These and other possible input variables are discussed in the next section.

5.1.3. Input variables

In the previous sections the input variables of Fancello et al.[13] and Parolas[58] were identified. These input variables are discussed first and why some are or are not used in the research. Unfortunately not all the variables are explicitly discussed in both researches, so the discussion may be biased by the interpretation of the researchers of the variables. After this discussion these variables are supplemented with other variables which may be important in the research.

Name of the vessel The name of a vessel is a unique identifier for a vessel. With the use of an unique identifier, behavior that is specific for a certain vessel can be taken into account. However as shown in Subsection 3.2.2 the name of a vessel is error prone. Therefore another unique identifier for the vessel is used.

Port of departure The port of departure might provide some information regarding the distance a vessel needs to cover, but the bigger the distance the bigger the uncertainty about an ETA. Therefore the port of departure is not included, but in stead use real-time geographical information.

Number of dockers required for loading This variable is not explained by Fancello et al. so it is unclear if this concerns dockers at the port of arrival or the port of departure. So his variable is not included in the research.

Longitude This variable provides the longitudinal position of a vessel. The values are positive for vessels on the eastern hemisphere with a maximum of 180 and negative for vessels on the western hemisphere with a minimum of -180. In combination with the latitudinal position this provides the location on the globe of a vessel.

Latitude This variable provides the latitudinal position of a vessel. The values are positive for vessels on the northern hemisphere with a maximum of 90 and negative for vessels on the southern hemisphere with a minimum of -90. In combination with the longitudinal position this provides the location on the globe of a vessel.

Distance to be covered This variable shows the distance a vessel needs to cover before the vessel arrives at the Port of Rotterdam. This variable does not provide additional information since the position of the vessel is already known and within this knowledge the distance to be covered is already enclosed. So distance to be covered is not used.

Current speed The current speed can be used to calculate the time it takes for a vessel to cover the remaining distance to the Port of Rotterdam. This speed does not remain constant and future speeds of the vessel are hard to predict.

Change in speed over the last 3 hours This variable indicates if a vessel is changing its speed and thus if adjustments need to be made to the ETA based on these changes.

Average speed based on last 12 hours This variable shows the average speed of the vessel of the last 12 hours. This provides an indication if the vessel has changed his speed a lot over the last few hours.

Time used for calculating the average speed Not every average speed is calculated over a time span of 12 hours. So this variable indicates how trustworthy the average speed is[58].

Length and breadth of the vessel The dimension of the vessel influence the requirements for the berth and handling equipment and thus may have an influence on for example waiting times. Also the bigger a vessel the more containers need to be loaded and offloaded so this also takes more time. Furthermore bigger vessels might get priority because they offer chances for bigger revenues.

ETA of the vessel Shippers try to arrive at the provided ETA and they adjust their speed and behavior to this ETA, so its an important input variable[58].

Now the input variables of Fancello et al. and Parolas are discussed, some variables are introduced that may be of importance for predicting the ETA of vessels en route to the Port of Rotterdam.

IMO number As already discussed an unique identifier may capture behavior related to a specific vessel. Since IMO is the least error prone unique identifier the IMO number is used as input variable.

Shipping line As discussed in Section 2.1, shipping lines invest in terminals so which terminals and even ports are visited by a vessel may depend on the shipping line. It could be that vessels of a shipping line have priority over other vessels so this may influence waiting times and thus the travel time.

Draught of the vessel The draught of a vessel may limit what ports it can visit and thus influence the route of a vessel. So this variable may also be needed to make accurate route predictions.

Navigational status of the vessel vessels communicate their status via AIS. Their status could be 0 which means sailing or 5 which means moored. A lot of other statuses are possible. The status gives an indication if the vessel is moving or waiting so it might influence the travel time of a vessel.

In conclusion an overview is provided of the variables that may have an influence on the ETA of the vessel.

- Latitude
- Longitude
- Current speed
- Change in speed over the last 3 hours
- Average speed based on last 12 hours
- Observations used for calculating the average speed
- Length and breadth of the vessel
- ETA provided by the vessel
- IMO number
- Shipping line
- Draught
- Navigational status

5.2. Possible data quality issues

In Subsection 3.2.2 possible problems with AIS messages are discussed. In this section these problems, what the consequences are for the dataset and how to handle these problems is discussed.

Unique identifiers Vessels might transmit wrong unique identifiers, so these identifiers are cross-checked with other unique identifiers to validate the number of vessels in the dataset and also to be sure the right unique identifier is used to identify vessels.

Vessel Type The vessel type is a variable that could be kept too general, wrongly inputted or not inputted at all. Since the dataset needs to consist of only container vessels, the vessel type is cross-referenced with another information source.

Length and beam Errors in length and beam consist of not displaying the information, displaying incorrect information or incorrect correlation between length and beam. As is shown in Section 5.4 length and beam are not available in the dataset but need to be computed. So checking these variables is not required. However when calculating these variables, some checks need to be incorporated.

Position It is possible that vessels transmit positions that are outside of the possible range of $-180°$ to $180°$ for longitude or $-90°$ to $90°$ for latitude. So these variables are checked in the dataset.

Vessel navigation status The navigation status needs to be manually updated by the officer of the watch or the navigation officer. In the system in 2007 crosschecking was not incorporated in AIS, causing vessel to display the incorrect status because updates were not entered. So the status of the vessels is cross-checked with its speed and position.

Draught In the researches of Bailey[4] and Harati-Mokhtari et al.[20] it was found that vessels did not report a draught or a draught of 0 meter. Also a few cases were found where the draught was off by several meters, which is a big difference. And might be the difference between grounding or not grounding a vessel. The draught of every vessel can not be checked since the data is historical, however vessels are checked to report a draught.

Destination About half of the vessels in the research of Harati-Mokhtari et al.[20] displayed wrong information in this field. Possible errors in the destination field are[4, 12, 20]:

- A number instead of a destination
- A country instead of a port
- Abbreviated names
- Showing not available, not defined or NULL
- Fake input, e.g. to hell
- No input
- Previous port as destination

Therefore the destinations are standardized in the dataset, so the destination can be used in the route identification.

ETA Problems with the ETA are not updated ETA's, ETA's in the past or ETA's in a very distant future. So the ETA's in the dataset are checked.

Because of these possible issues, the required quality standards are not met, because data is needed that is accurate, complete and consistent. Therefore a workflow is constructed to improve the data quality of AIS messages, as shown in Figure 5.1. This workflow is discussed in the remainder of the chapter. The dataset is gathered and explored to identify possible problems. The dataset is cleaned and manipulated to be able to use the data. After the cleaning and manipulation the data is explored again to be sure all problems have been solved.

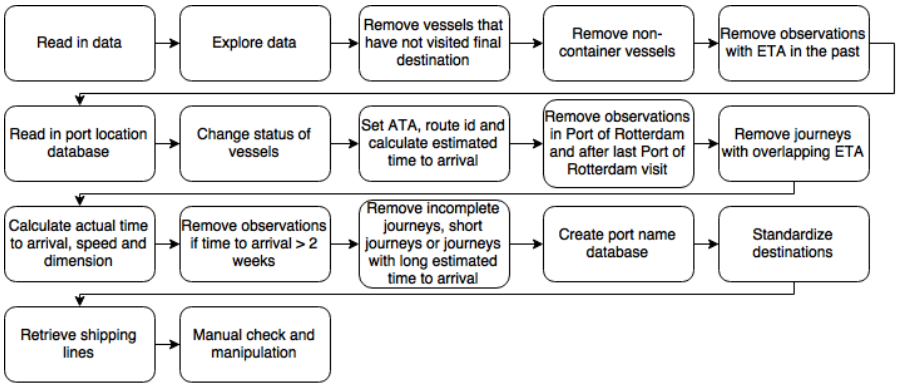


Figure 5.1: Workflow to improve the quality of an AIS dataset.

5.3. Data collection

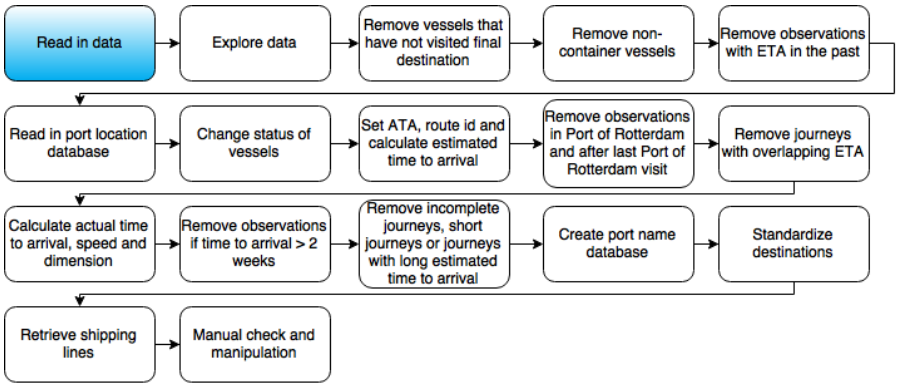


Figure 5.2: Collection step of framework to improve the quality of an AIS dataset

Figure 5.2 shows the collection step op the proposed framework. The dataset for the research is probably created by Transsis and is provided by TNO. The dataset consists of 5,95 million AIS messages from multiple vessels spanning the period April 1st, 2014 to October 31st, 2015. So the dataset covers roughly 1,5 years. Since documentation is not present with the dataset, it is unclear if the dataset is actually created by Transis. Furthermore a dataset is known, created by Transis, where the ETA is not from the AIS messages of a vessel but communicated by the shippers agent. So it is also unclear if the ETA is from the AIS message or not. The input variables identified in the previous section are all present in the dataset as is showed in Section 5.4 or can be computed from the dataset as is showed in Section 5.6. After collecting the data, the dataset is read into R for the research.

5.4. Data exploration

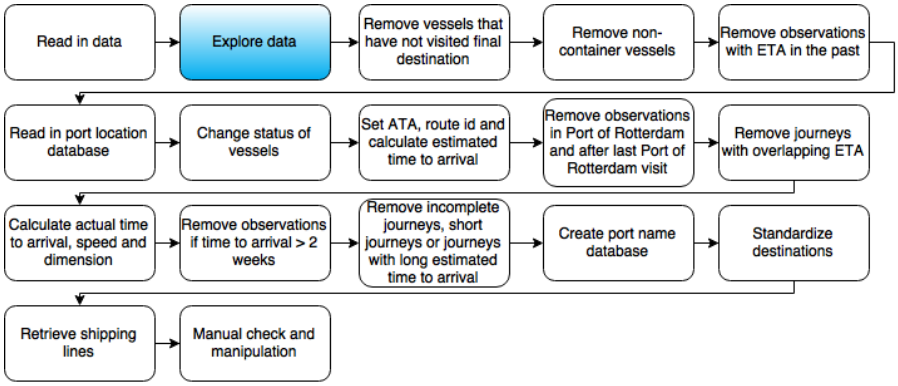
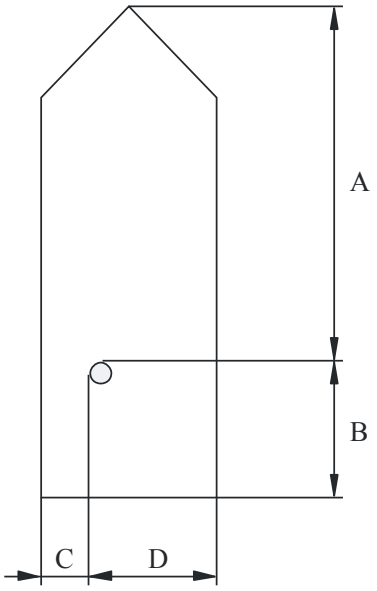


Figure 5.3: Exploration step of framework to improve the quality of an AIS dataset

Figure 5.3 shows the exploration step op the proposed framework. Before the data is used, an understanding of the data is generated. The data is explored. To get an idea about the structure of the data first glimpse from the dplyr package is used. The results are shown in Output 5.1. From these results is concluded that the dataset has almost 6 million observations with 20 variables. The variables are shortly discussed in Table 5.3.

Output 5.1: Output of the Glimpse function. First line shows the number of observations and the second line the number of variables. Every following line shows a variable, its type and the first few instances of that variable.

```
Observations: 5,951,303
Variables: 20
$ timestamp   (time) 2014-04-01, 2014-04-01, 2014-04-01,...
$ mmsi        (int) 209467000, 210001000, 212706000,...
$ status      (int) 0, 5, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0,...
$ speed       (dbl) 9.2, 6.7, 16.3, 17.8, 17.3, 6.7,...
$ latitude    (dbl) 53.560340, 51.982980, 55.074160,...
$ longitude   (dbl) 9.765097, 4.090978, 14.191810, 10.7...
$ course      (int) 282, 292, 40, 106, 277, 219, 198,...
$ heading     (int) 282, 291, 40, 104, 276, 218, 198,...
$ destination (chr) NA, "KLAIPEDA", "SAINT PETERSBURG",...
$ draught     (int) 74, 61, 91, 144, 149, 70, 101, 23,...
$ eta         (time) 2014-10-04 21:59:00, 2014-10-02 10...
$ imo         (int) 9483671, 9162681, 9386718, 9501368,...
$ name        (chr) "NORDIC STANI", "ALASA",...
$ callsign    (chr) "5BMB3", "C4RW2", "5BTT2", "DJBF2",...
$ type        (chr) "container ship (fully cellular)",...
$ ais_type    (int) 71, 70, 71, 70, 70, 74, 70, 52, 71,...
$ bow         (int) 142, 110, 155, 141, 141, 108, 220,...
$ stern       (int) 9, 12, 13, 225, 225, 12, 73, 8, 224,...
$ port        (int) 12, 11, 13, 29, 29, 10, 28, 2, 29,...
$ starboard   (int) 11, 7, 13, 19, 19, 8, 12, 7, 19, 19,...
```



	Number of bits	Bit fields	Distance (m)
A	9	Bit 21-Bit 29	0-511 511 = 511 m or greater
B	9	Bit 12-Bit 20	0-511 511 = 511 m or greater
C	6	Bit 6-Bit 11	0-63; 63 = 63 m or greater
D	6	Bit 0-Bit 5	0-63; 63 = 63 m or greater

The dimension A should be in the direction of the transmitted heading information (bow)
Reference point of reported position not available, but dimensions of ship are available: A = C = 0 and B ≠0 and D ≠0.
Neither reference point of reported position nor dimensions of ship available; A = B = C = D = 0 (= default).
For use in the message table, A = most significant field, D = least significant field.

M.1371-41

Figure 5.4: Overview how to construct vessel dimensions. Image from International Telecommunication Union[28]

Table 5.3: AIS variables in the dataset, explanation retrieved from[28, 49, 50]

AIS Variable	Explanation
timestamp	UTC time when the data was generated.
mmsi	The MMSI number of the vessel.
status	Communicates the navigational status of the vessel. 0 = under way using engine, 1 = at anchor, 2 = not under command, 3 = restricted maneuverability, 4 = constrained by her draught, 5 = moored, 6 = aground, 7 = engaged in fishing, 8 = under way sailing, 9 = reserved for future amendment of navigational status for vessels carrying DG, HS, or MP, or IMO hazard or pollutant category C, high speed craft (HSC), 10 = reserved for future amendment of navigational status for vessels carrying dangerous goods (DG), harmful substances (HS) or marine pollutants (MP), or IMO hazard or pollutant category A, wing in ground (WIG); 11 = power-driven vessel towing astern (regional use); 12 = power-driven vessel pushing ahead or towing alongside (regional use); 13 = reserved for future use, 14 = AIS-SART (active), MOB-AIS, EPIRB-AIS 15 = undefined = default (also used by AIS-SART, MOB-AIS and EPIRB-AIS under test)
speed	Shows the speed over the ground in km/h
latitude	The latitudinal position of a vessel. The value lies between $-90°$ and $90°$. Negative for the western hemisphere and positive for the eastern hemisphere.
longitude	The longitudinal position of a vessel. The value lies between $-180°$ and $180°$. Negative for the southern hemisphere and positive for the northern hemisphere.
course	Course over ground in $1/10°$ = (0-3599). 3600 or (E10h) = not available = default. 3 601-4 095 should not be used. This position is relative to true north: $0.1\,\text{degree}$
heading	The heading of the vessel in degrees (0 - 359). 511 is default, indicates not available
destination	Indicates the next destination of the vessel.
draught	The draught of the vessel in 1\10 meters. So 255 is 25.5m. 255 indicates a value of 25.5m or more.
eta	Estimated time of arrival of the vessel in UTC. The timestamp is formatted in Year-Month-Day Hour:Minute:Second.
imo	The IMO number of the vessel. 0 = not available = default – Not applicable to SAR aircraft 0000000001-0000999999 not used 0001000000-0009999999 = valid IMO number; 0010000000-1073741823 = official flag state number.
name	The name of the vessel.
callsign	The callsign of the vessel.
type	The vessel type.
ais_type	Numeric value for the ais type
bow	Distance from AIS system to bow in meters
stern	Distance from AIS system to stern in meters
port	Distance from AIS system to port in meters
starboard	Distance from AIS system to starboard in meters

The bow, stern, port and starboard variables need some extra explanation on how to calculate the dimensions of the vessel and when these dimensions can be calculated. This is given in Figure 5.4.

With al the variables provided in the dataset the input variables as identified in Subsection 5.1.3 can be calculated. Data exploration is continued to check if the data is tidy. Tidy data is introduced by Hadley Wickham and has three characteristics[88]:

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

To check if these characteristics hold the first 10 entries of the dataset are observed. Due to space restrictions the output is not shown. For this dataset the characteristics hold so the dataset does not need to be transformed. To gain some more insight about the data in the dataset the summary function in R is used. The output is shown in Output 5.2

Output 5.2: Output of the summary function. For every variable the minimal and maximum values are shown. Furthermore it shows the mean, median and values at the 1st and 3rd quartile

```
  timestamp                      mmsi             status
Min.   :2014-04-01 00:00:00   Min.   :209165000   Min.   : 0.000
1st Qu.:2014-09-06 16:18:00   1st Qu.:236262000   1st Qu.: 0.000
Median :2015-02-06 09:02:00   Median :311257000   Median : 0.000
Mean   :2015-01-28 19:57:09   Mean   :368816675   Mean   : 1.883
3rd Qu.:2015-06-23 11:02:00   3rd Qu.:477519400   3rd Qu.: 5.000
Max.   :2015-10-31 23:58:00   Max.   :667001412   Max.   :97.000

     speed            latitude
 Min.   :  0.000   Min.   :-71.01
 1st Qu.:  0.000   1st Qu.: 30.99
 Median :  6.900   Median : 45.61
 Mean   :  7.674   Mean   : 38.74
 3rd Qu.: 15.200   3rd Qu.: 53.34
 Max.   :232.000   Max.   : 82.09

  longitude           course         heading
Min.   :-179.992   Min.   :  0.0   Min.   : -1.0
1st Qu.:  -2.478   1st Qu.: 83.0   1st Qu.: 90.0
Median :   5.211   Median :187.0   Median :196.0
Mean   :  14.208   Mean   :178.3   Mean   :188.1
3rd Qu.:  25.379   3rd Qu.:269.0   3rd Qu.:279.0
Max.   : 179.999   Max.   :404.0   Max.   :511.0
                                   NA     :4351

destination          draught
Length:5951303     Min.   :  1.00
Class :character   1st Qu.: 72.00
Mode  :character   Median : 97.00
                   Mean   : 96.68
                   3rd Qu.:122.00
                   Max.   :255.00
                   NA     :97567

     eta                           imo              name
Min.   :1900-01-01 00:00:00   Min.   :8100636   Length:5951303
1st Qu.:2014-10-06 15:00:00   1st Qu.:9251377   Class :character
Median :2015-02-02 02:00:00   Median :9344722   Mode  :character
Mean   :2014-07-23 22:30:06   Mean   :9343782
3rd Qu.:2015-06-26 10:00:00   3rd Qu.:9466245
Max.   :2016-03-20 17:45:00   Max.   :9713349
NA     :84436

  callsign
Length:5951303
Class :character
Mode  :character
```

```
     type              ais_type          bow              stern
Length:5951303     Min.   : 0.00   Min.   :  0.0   Min.   :  0.00
Class :character   1st Qu.:70.00   1st Qu.:126.0   1st Qu.: 14.00
Mode  :character   Median :71.00   Median :144.0   Median : 70.00
                   Mean   :70.47   Mean   :153.7   Mean   : 88.76
                   3rd Qu.:71.00   3rd Qu.:201.0   3rd Qu.:122.00
                   Max.   :94.00   Max.   :276.0   Max.   :386.00

      port          starboard
Min.   : 0.00   Min.   : 0.00
1st Qu.:11.00   1st Qu.:10.00
Median :16.00   Median :16.00
Mean   :16.58   Mean   :16.86
3rd Qu.:21.00   3rd Qu.:23.00
Max.   :41.00   Max.   :40.00
```

Every attribute of the dataset, that at this first glance display values that are not possible, are discussed.

AIS status The range of AIS statuses is from 0 to 15. However a maximum of 97 is shown so statuses that are not possible are reported in the dataset. Before doing predictions it is ensured that these statuses are changed or not in the dataset.

Course The course of a vessel can lie between 0 and 360, however a maximum of 404 is reported. 404 means that no course information is available so it might be necessary to manipulate these inputs.

Heading As with course the heading has a range between 0 and 360. vessels report a heading of -1 or above 360. Furthermore for about 4000 messages the heading is NA(not available). Since heading and course are related heading is not used for the predictions.

Draught About 100.000 observations have a draught of NA, so it might be necessary to calculate draughts for vessels in the data manipulation.

ETA In the ETA attribute almost 85.000 observations do not report an ETA, also ETA's for January 1st, 1900 are reported. Since time travelling is still imposible, these ETA's are an error and voyages containing these ETA's are not used.

Position data The positions reported by the vessels are all within the possible ranges. However in order to check if every position location is valid all the data-points are mapped on the world map. The results are shown in Figure 5.5. Some data points occur in the middle of Asia and North America. These data points need to be taken care of in the data cleaning and manipulation. Furthermore some more points of interest can be found in the vicinity of Antartica, since no other data-points occur in the vicinity these may be faulty positions.

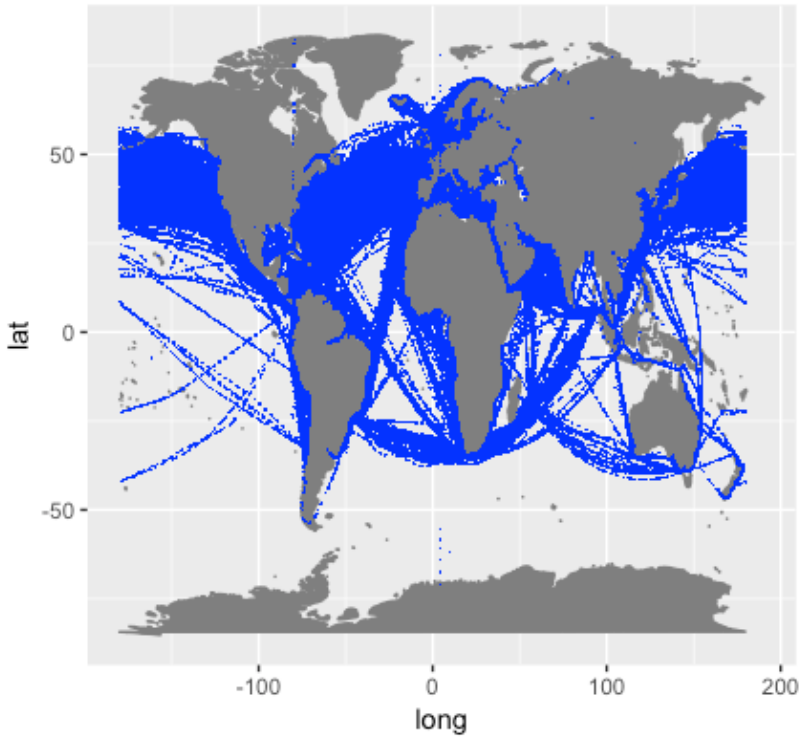


Figure 5.5: Overview of all the data-points on the world map

Since some variables are of the type character no information about their values is presented. Therefore variables that according to Subsection 3.2.2 are error prone are checked. In the exploration phase the only concern is if the dataset consists of only container vessels therefore all the unique values in the vessel type attribute are shown in Output 5.3.

Output 5.3: Output showing the unique vessel types in the dataset.

```
 [1] "container ship (fully cellular)" "general cargo ship"
 [3] "utility vessel"                  "tug"
 [5] "limestone carrier"               "container ship"
 [7] "passenger/ro-ro ship (vehicles)" "general cargo s"
 [9] "chemical/products tanker"        "livestock carrier"
[11] "trailing suction hopper dredger" "heavy load carrier"
```

It turns out that also other vessel types are in the dataset. However the vessel types are error prone. Therefore the type of every vessels is validated using the database of the International Maritime Organization[1]. According to this database every vessel type as communicated via AIS is correct so messages from vessels that are not a container vessel are removed during the data cleaning.

[1] https://gisis.imo.org/Public/SHIPS/Default.aspx

In Subsection 3.2.2 problems that may occur in unique identifiers are identified, therefore the amount of unique MMSI and IMO number are compared. If they are not equal this means that one of the two contains errors and can not be used for predictions. MMSI and IMO should be equal because both are unique identifiers but are respectively 815 and 733. Both variables do not have missing values, so a check for default values is performed. The default value for MMSI is 1193046[73]. This number is lower than the minimal value in the dataset. So no defaults for MMSI are present in our dataset. The default value for IMO is 303174162[14]. This number is also not present. So the MMSI and IMO number are double-checked with the unique names in the dataset. IMO and names are the same amount, 733, so the problem probably lies with the MMSI. To check this a table is compiled with every combination of MMSI, IMO and name. So a check can be performed if a vessel transmits multiple MMSI values. It turned out that some vessels transmit multiple values for MMSI, these vessels are shown in Table B.1. 76 vessels have an MMSI number that changed, some even multiple times. This accounts for the difference. In the remainder of the thesis IMO numbers are used to identify vessels.

At this phase the other attribute are not checked because these are related to a specific observation and checking every observation will be very time consuming.

5.5. Data Cleaning

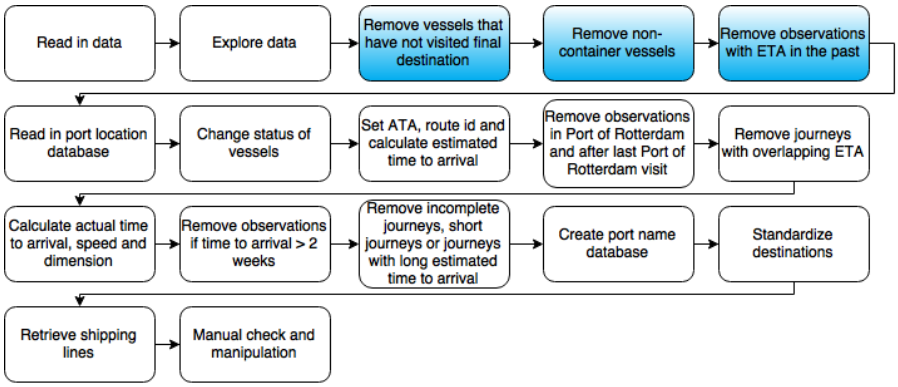


Figure 5.6: Cleaning steps of framework to improve the quality of an AIS dataset

In this section the cleaning steps of the framework are discussed. Figure 5.6 shows the cleaning steps op the proposed framework. First vessels that have not visited the Port of Rotterdam are removed since these are outside the scope of this research. The entrance to the Port of Rotterdam lies on the line between NB 51.976510 and 51.986627 and OL 4.072472 and 4.083678 according to Google maps. So all the vessels that passed this line are identified. To be sure that not a single vessel is missed due to measure inaccuracies, all the vessels that pass the area covered by NB 51.95 and 52.0 and OL 4.05 and 4.10 are identified. This area is shown in Figure 5.7. Because only vessels that enter the Port of Rotterdam are of interest, their course needs to be between 0 and 180. Otherwise the vessel is leaving the harbor. A vector is created with every IMO number that enters the Port of Rotterdam. After the first check based on passing the defined area at the entry of the port all the remaining vessels are manually checked. This is done by taking an anti-join of the original dataset and the new dataset so only the remaining vessels will be under consideration.

(a) Entry to port of Rotterdam

(b) Entry to port of Rotterdam up close

Figure 5.7: Defined entry area to the Port of Rotterdam

The dataset is first searched using the term ”Rotte” since some vessels made an typo and stated Rottep or Rottesdam for example. Using this search term two vessels are identified that stated Rotterdam as their destination, however they are already in the port of Rotterdam and leave the port without returning to Rotterdam. So these should not be added to the dataset. Also 3 vessels are identified that were en route to Rotterdam but did not reach the port within the timespan of the dataset. So these are also not added to the dataset. One vessel is identified that suddenly occurred in the Port of Rotterdam after not broadcasting AIS messages for a period of 3 months. So also this vessel is not included in the dataset. 24 vessels are found that did visit the Port of Rotterdam. These were not included because they travelled through the defined area within the hour and thus were missed. Or the vessels visit the ”Maasvlakte 2” and their course was already bigger than 180 because they already turned into ”Maasvlakte 2”. These vessels are added to the dataset and another anti-join is performed to look for other vessels that do visit the port but are not in the cleaned dataset. The dataset is manually filtered to a larger area at the entry point to see what vessels did also pass. Some vessels are discovered that did not state the Port of Rotterdam as a destination but did visit the port and some vessels that used abbreviations in the destination so were missed by the previous query. These are also added to the dataset.

A vector is created containing all the manually identified vessels. This vector is used to add all these vessels to the dataset. When all the vessels are added, the dataset is cleaned based on their vessel type. Furthermore the dataset is filtered for messages that have a positive ETA, so their ETA lies after 01-01-1970 00:00. The steps as discussed are shown in Algorithm 5.1.

Algorithm 5.1 Clean vessels

1: Save every unique IMO number that enters Port of Rotterdam
2: Filter dataset on saved IMO numbers
3: Construct vector with IMO number that needs to be added to dataset
4: Filter wrongly excluded vessels from dataset and add to new dataset
5: Filter on vessel type
6: Filter on ETA

Output: ERP_AIS_clean

Now only container vessels that did visit Rotterdam are in the dataset. A new plot is made of all the data-points on the world map to identify if outliers are still present. Outliers are still present in the dataset, for example data-points in the center of America. We can see these outliers in Figure 5.8.

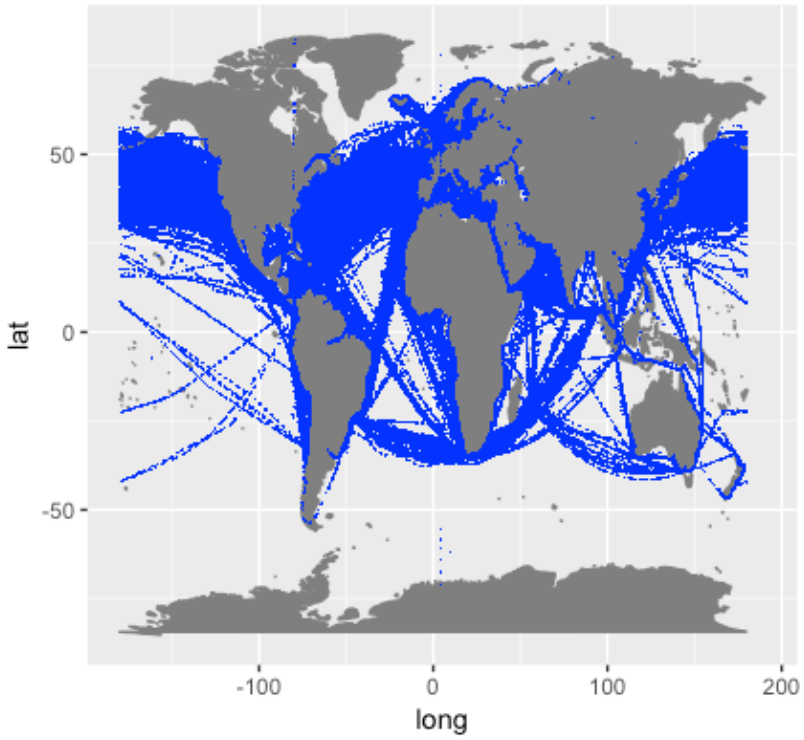


Figure 5.8: Overview of all the data-points after cleaning on the world map

5.6. Data manipulation

After the data cleaning step a dataset that consists of container vessels that visit the Port of Rotterdam is obtained. Now a clean dataset regarding the vessels is obtained, variables that may contain errors are manipulated and also observations are removed if they contain errors that can not be solved.

In this section the manipulation part of the framework to improve the data quality of the AIS database is presented. The steps of the framework are presented in Figure 5.9 and every step will be discussed in a subsection of this section. The entire code can be found in Appendix C. Although the code is tailored to the dataset as provided by TNO, some key steps to improve the data quality of a database consisting of decoded AIS messages are identified.

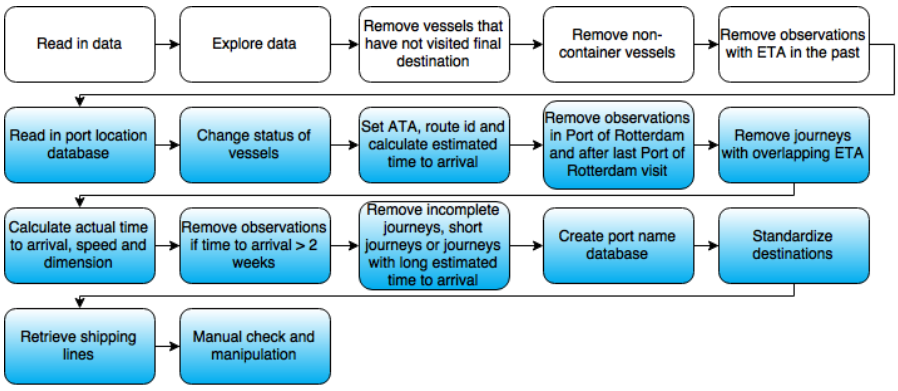


Figure 5.9: Manipulation steps of framework to improve the quality of an AIS dataset

5.6.1. Port location database

The first step is reading in a database that contains all the ports worldwide and location information regarding these ports. The database can be found online[2]. The MS access database is downloaded and converted to CSV. This CSV file is read into the workspace. The location info is manipulated so it is represented in the same notation as in the AIS data as shown in Algorithm 5.2. This database is used to manipulate attributes based on location data of the ports.

Algorithm 5.2 Read Port location database

```
Read in csv file
Create longitude and latitude column
for all Ports do
    Manipulate longitude and latitude notation
end for
```
Output: WPI

5.6.2. Navigational status

Based on the errors as discussed in the literature review the navigational status of a vessel is checked to be correct related to its location. The research only utilizes when a vessel is in a port, so status equals 5, and when a vessel is not in a port, so status equals not 5. The code in Algorithm 5.3 is used to change the status of a vessel when needed. If a vessel is in a port being (off)loaded its speed will be at or below 0.5 km/h. If a vessel is traveling at a speed at or below 0.5 km/h and its status is 0 while near a port the status is changed to 5. It is also possible that a vessel leaves a port but forget to set its navigational status to sailing (0). Therefore if the vessel travels at a speed above 0.5 km/h with status moored (5), the status is changed to sailing. An example is shown in Table 5.4.

Algorithm 5.3 Change status

Input: ERP_AIS_clean
```
for all observations do
    Read observation
    if status = sailing and speed <= 0.5 and vessel is near port then
        Change status to moored
    else if speed > 0.5 and status = moored then
        Change status to sailing
    end if
end for
```
Output: ERP_AIS_clean$status

Table 5.4: Example of statuses. First four rows show original statuses, next rows show manipulated statuses

IMO	Speed	Status	Near port
1	0.0	0	Yes
1	0.0	5	Yes
1	0.0	0	No
1	1.5	5	Yes
1	0.0	5	Yes
1	0.0	5	Yes
1	0.0	0	No
1	1.5	0	Yes

[2] http://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true&_pageLabel=msi_portal_page_62&pubCode=0015

After this manipulation every vessel that is in a port being (off)loaded has status 5 and vessels that are traveling have a status that is not 5. So the status message is used to identify if a vessel is in a port.

5.6.3. Actual Time of Arrival, route ID and estimated time to arrival

Since the time of arrival of vessels in the Port of Rotterdam is the target value, information regarding the Actual Time of Arrival (ATA) is needed. Also each voyage of a vessel is differentiated, a route ID is assigned to every voyage of the vessel. Also according to the research of Parolas the ETA as communicated via the AIS messages is important in predicting the ETA of a vessel[58]. Since data can not be generalized the ETA is transformed into a variable that can be used. An ETA is dependent on the expected travel time of a vessel and the current time. Since travel times can be generalized the expected travel times are calculated based on the communicated ETA. Since the ETA changes, if done correctly, for every destination along a voyage the ETA of the current observation can not be used. Also since it is unknown for which destination an ETA is communicated, the ETA is used that is communicated when the vessel is first moored in the Port of Rotterdam, since it is most probable that this ETA is related to Rotterdam.

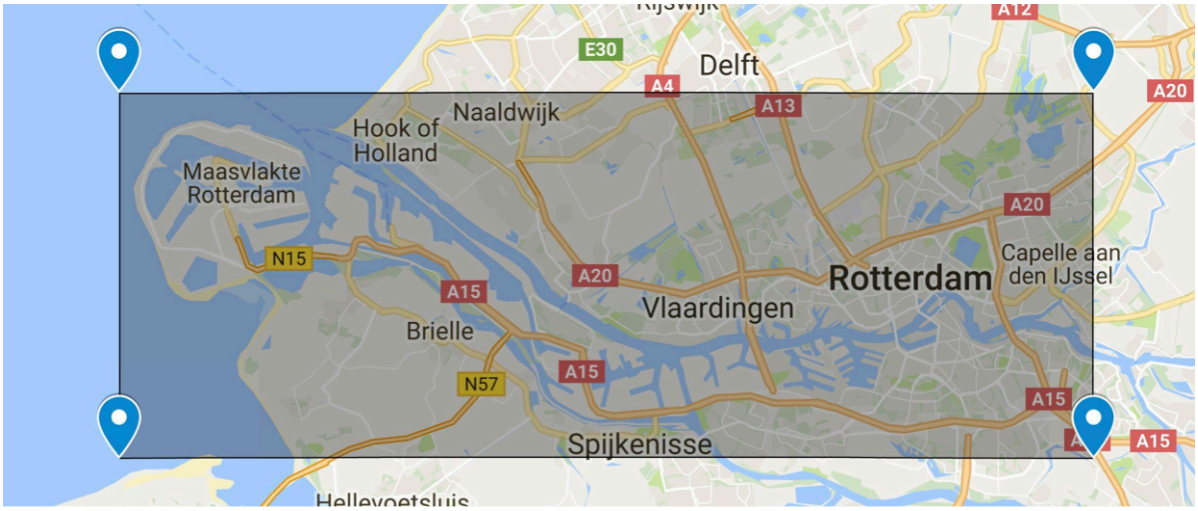


Figure 5.10: Area defined as the Port of Rotterdam in which vessels are checked.

The R code does the following. It first orders the dataset by IMO number and timestamp so all the observations for a vessel are grouped and each observation for a specific vessel is a point later in time then the previous observation for that vessel. The attributes ATA, route ID and estimated travel time are added to the dataset with value Not Available(NA). The variable that counts the amount of routes for a vessel is set to 1 and the dataset is looped over. The first if statement checks if a new vessel is encountered and resets the route counter to 1 if needed. The second if statement checks if the vessel is in the area of the Port of Rotterdam, as defined in Figure 5.10, with status moored. The while loop pauses at the first observation of a moored vessel in the Port of Rotterdam. The timestamp of this observation is used as the ATA of the vessel. Furthermore the route ID is assigned to this observation and the estimated travel time is calculated in hours based on the current ETA and the current timestamp.

The next step in the algorithm is a while loop that loops back through the dataset while the IMO number stays the same and the ATA is not set. For every observation it passes it sets the ATA to the ATA of the first observation while moored in the Port of Rotterdam, gives the observation the corresponding route ID and it calculates the estimated travel time in hours based on the ETA of the first observation while moored in the Port of Rotterdam and the current timestamp. When this loop finished all the attributes for that voyage until the vessel becomes moored in the Port of Rotterdam have been set accordingly. Then another while loop goes forward into the dataset while the vessel remains moored in the Port of Rotterdam. Since the behavior of the vessel in the Port of Rotterdam is not of interest, the ATA is set to 1 to ensure that the previous while loop not also goes through these observations and to be able to remove these observations in a later stage. The counter of the original while loop is set to the row number of the first observation that has not yet been observed and the route counter is incremented by 1. The function then searches for the next stop in Rotterdam. These steps are all repeated until the entire dataset has been looped over, this is shown in Algorithm 5.4. When the function finishes the class of the ATA attribute is set to POSIXct so that it will display dates and not a numeric representation of the amount of seconds that has passed since 01-01-1970 00:00. An example is shown in Table 5.5

Algorithm 5.4 Set ATA, Route ID and calculate estimated time to arrival

Input: ERP_AIS_clean
1: Sort by IMO and timestamp
2: Create columns ATA, route_id and est_traveltime
3: route ← 1
4: max ← number of rows in ERP_AIS_clean
5: i ← 1
6: **while** i ≤ max **do**
7: Read ith observation
8: **if** new IMO **then**
9: route ← 1
10: **end if**
11: **if** Vessel is moored in Port of Rotterdam **then**
12: Set ATA to timestamp
13: Set route ID
14: Calculate estimated time to arrival
15: **while** Vessel is traveling to Port of Rotterdam **do**
16: Set ATA and route ID
17: Calculate estimated time to arrival
18: **end while**
19: **while** Vessel stays moored in Port of Rotterdam **do**
20: Set ATA to 1
21: **end while**
22: increment i to first unchanged observation
23: increment route by 1
24: **else**
25: increment i by 1
26: **end if**
27: **end while**
Output: ERP_AIS_clean

After these manipulations the amount of observations in our dataset is reduced. Since some vessels do not return to Rotterdam after a visit in the scope of this dataset all the observations for a vessel after the last visit to the Port of Rotterdam are removed. Also since the behavior of a vessel inside the Port of Rotterdam when it is moored is not of interest these observations are removed.

Table 5.5: Example of manipulated messages. First rows show original statuses, next rows show manipulated statuses. The fifth message will be deleted

IMO	Timestamp	ETA	Status	At dest	Route ID	ATA	Est time to arrival
1	30-05 09:00	31-05 12:00	0	No			
1	30-05 13:00	31-05 12:00	5	No			
1	31-05 08:00	31-05 12:00	0	No			
1	31-05 11:00	31-05 12:00	5	Yes			
1	31-05 12:00	31-05 12:00	5	Yes			
1	31-05 18:00	02-06 12:00	0	No			
1	30-05 09:00	31-05 12:00	0	No	1	31-05 11:00	27 hours
1	30-05 13:00	31-05 12:00	5	No	1	31-05 11:00	23 hours
1	31-05 08:00	31-05 12:00	0	No	1	31-05 11:00	4 hours
1	31-05 11:00	31-05 12:00	5	Yes	1	31-05 11:00	1 hour
1	31-05 12:00	31-05 12:00	5	Yes	-	01-01-1970 00:00:01	-
1	31-05 18:00	02-06 12:00	0	No	2	01-06 13:00	42 hours

5.6.4. Overlapping ETA

Now the ATA for every voyage is known, some checks are performed on the ETA since the communicated ETA's are error prone. ETA's may not be changed for the next destination or voyage and thus span multiple journeys. Also instances are discovered where the ETA was communicated for voyages that had not yet occurred. A small example is presented in Table 5.6. As becomes clear from this example the ETA that is communicated is most probably related to the voyage with route ID 4. Although manipulating these ETA's is preferred, captain estimates are hard to compute so these overlapping ETA's are removed. These overlapping ETA's are removed because they influence the estimated travel time and thus one of the possible input variables for prediction.

Table 5.6: Example of false ETA's

IMO	Route ID	ETA	ATA
1	1	31-05-2017 12:00	30-04-2017 15:21
1	2	31-05-2017 12:00	09-05-2017 16:41
1	3	31-05-2017 12:00	21-05-2017 00:20
1	4	31-05-2017 12:00	31-05-2017 13:15
1	5	31-05-2017 12:00	07-06-2017 04:54
1	6	31-05-2017 12:00	30-08-2017 07:49

Overlapping ETA's are identified by looping over the dataset. If the IMO of the current observation and the next observation are equal but the route ID is not two different voyages for the same vessel are identified. The current observation is the last observation of the first voyage and thus this observation also contains the ETA for that voyage. In another while loop the last observation of the next voyage is identified and the corresponding ETA is compared to the ETA of the first voyage. When these ETA's are the same the voyage with the least accurate ETA is identified and the IMO and route ID of that voyage is stored. These steps are repeated until all successive voyages for every vessel are checked. Next the resulting data frame is used to set the IMO of observations, that have overlapping ETA, to NA and remove all observations where the IMO is NA. A pseudocode is shown in Algorithm 5.5.

Algorithm 5.5 Remove overlapping ETA

Input: ERP_AIS_relevant
```
i ← 1
Create data frame remove
while i < number of rows in ERP_AIS_relevant do
    Read ith observation
    if i+1th observation is new journey of same vessel then
        Search last observation of next journey
        if ETA are equal then
            Add IMO number and route ID of journey with least accurate ETA to remove
        end if
        Set i to last observation of last observed journey
    else
        Increment i by 1
    end if
end while
Set IMO for every observation where IMO number and route ID is in remove to NA
Remove all observations with IMO = NA
```
Output: ERP_AIS_relevant

5.6.5. Actual time to arrival, dimensions and speed

The dataset now consists of container vessels that visited the Port of Rotterdam with an accurate ETA. Since only vessels with a travel time that is below a threshold are relevant to the research the travel time of every observation is calculated. When an observation is relevant to the research the dimensions of the vessel are calculated and the speed variables that have an influence on the predictions according to Parolas are calculated.

First a travel time threshold of 2 weeks is set. The research of Parolas concluded that a timeframe of 1 week would be sufficient for the stakeholders in the Port of Rotterdam[58]. However a bigger timeframe is taken since vessels are included that visit multiple ports. Then the attributes avg_speed(the average speed), obs_speed (number of observations to calculate the average speed), d_speed (the change in speed over the last 3 hours), length (the length of the svessel, width(the width of the svessel and travel_time (the actual time to arrival for the vessel) are added to the dataset and set to NA. The for-loop manipulates every observation. First the actual time to arrival is calculated. Further calculations are only performed if the actual time to arrival from the point of the observation lies within the travel time threshold. First if the bow attribute is set the length and width of the vessel can be calculated as shown in Figure 5.4. Then a variable avg is set to the current speed of the vessel and another for-loop is entered. When j equals 3 the speed difference over the last three hours is set. Furthermore the variable avg is increased with the speed of previous observations until j equals 12 or another voyage is encountered. When j equals 12 or another voyage is encountered, the average speed is calculated according to the number of observations and the number of observation is stored in the obs_speed attribute. A pseudocode is shown in Algorithm 5.6 When the calculations are done observations are removed, that have a travel time that exceeds the threshold, to reduce the size of the dataset and because a bigger time horizon is irrelevant.

Algorithm 5.6 Calculate travel time, dimension and speed variables

Input: ERP_AIS_relevant, travel time threshold
1: Create columns: avg_speed, obs_speed, d_speed, length, width and travel_time
2: **for all** Observations in ERP_AIS_relevant **do**
3: travel_time ← Calculate actual time to arrival
4: **if** Actual time to arrival < travel time threshold **then**
5: Calculate dimensions and speed variables and store in corresponding column
6: **end if**
7: **end for**
Output: ERP_AIS_relevant

5.6.6. Incomplete\short routes and large estimated time to arrival

The dataset is reduced to a dataset that consists of information two weeks prior to arrival in the Port of Rotterdam for container vessels. Although AIS data is a reliable data source with a low latency a lot of missing observations are discovered in the dataset. Therefore the data is cleaned so it only consists of (near) complete routes that do not have large gaps. Algorithm 5.7 identifies the number of observation per voyage, the duration of the voyage, the difference between the number of observations, the percentage of the voyage that is covered by the observations and the biggest estimated travel time. Coverage is defined as the number of observations divided by the timespan of the voyage. So an voyage of 100 hours with 95 observations is covered for 95% and an voyage of 100 hours with 50 observations is covered for 50%. A route is considered complete when it is covered for at least 90%. So voyages that are below 90% are identified and stored in a data frame. Furthermore voyages that are very short and have less then 5 observations are identified and stored in the same data frame. The same holds for voyages with a maximum estimated travel time that is above 1000 hours or is negative. Because these ETA's are very likely wrong. The resulting data frame is used to remove voyages from the dataset that do not satisfy the mentioned requirements.

Algorithm 5.7 Remove incomplete or short journeys and journeys with long ETA

Input: ERP_AIS_relevant
1: Create data frame with statistics for every voyage
2: Identify voyages that do not meet requirements
3: Remove identified voyages
Output: ERP_AIS_relevant

Algorithm 5.8 identifies voyages with large gaps between observations. If a voyage has a gap that is larger then 5 hours the voyage is stored in a data frame and this data frame is used to remove all voyages that have a large gap.

Algorithm 5.8 Remove journeys with large gaps

Input: ERP_AIS_relevant
1: **for all** Observations **do**
2: Search gaps between observations
3: Identify and store voyages with gaps
4: **end for**
5: **if** Voyages with gaps in dataset **then**
6: Remove voyages with gaps
7: **end if**
Output: ERP_AIS_relevant

5.6.7. Standardize destinations

The dataset now describes the last two weeks of a complete voyage for container vessels to the Port of Rotterdam. In order to be able to use the routes of the vessel knowledge is needed about their destinations. However destination information is manually inputted and therefore is very error prone. The errors could be for instance typos or misleading destinations. In the dataset destinations like ”Verweggistan” or ”Going to hell” are discovered. Since the destination input is manual there are also a lot of different notations possible. For instance some use the port code to describe the destination, while others state the port of origin followed by the port of destination and also some add the country to the destination. The notation of the ports is standardized, to be able to solve errors in the stated destination. Also since a database with port codes is not available, such a database is created to translate port codes into port names. This will be the first step.

In order to construct a database two websites are scraped with R. `http://www.nslworld.net/ports.php` and `https://www.marinetraffic.com/en/ais/index/ports/all/per_page:50`. A data frame is created for each website. Because marinetraffic also stores information about anchorages and marinas, the data frame from marinetraffic is filtered. Also entries into the database that have no Port Code inputted are removed. Furthermore only the first three columns are selected because these are relevant for the database. The space in the Port Codes is removed and the order and the names of the columns are changed so they are the same as the data frame from nslworld. Both data frames are combined, every entry is transformed to uppercase and duplicate entries are removed from the database. In the final step two lists are constructed. One for Port Codes and one for Port Names. Both lists contain 26 lists, one for each letter of the alphabet. In these lists Port Codes or Port Names are stored that start with the corresponding letter. The algorithm is shown in Algorithm 5.9

Algorithm 5.9 Create database with port names and codes

1: ports1 ← Scrape nslworld
2: ports2 ← Scrape Marinetraffic
3: Manipulate ports2
4: ports ← Combine ports1 and ports2
5: Remove duplicates from ports
6: ports_codes ← Create list arranged alphabetically by port code
7: ports_names ← Create list arranged alphabetically by port name

Output: ports_codes, ports_names

Now a database of Port Codes and Port Names is obtained, the notation of the stated destinations is cleaned. Furthermore it is discovered in the dataset that due to the hourly sample rate some destinations are already changed for the first moored observation of a container vessel in a port. Therefore this is checked and the destination is manipulated if needed. To achieve this the dataset is looped over and for every new destination this destination is passed into a function. This function cleans the notation and will be discussed in a moment. The function returns a cleaned notation and this is stored in the observation. While the destination, vessel en voyage remains the same, the result of the function is stored in the following observations. Last it is checked if the next observation is the same vessel on the same voyage but with a status that has changed to moored. If this is the case and the destinations are not the same, the next observation already states the next destination of the vessel because the destination was changed between the 2 hourly samples. Therefore the destination of the next observation is changed. The pseudocode is shown in Algorithm 5.10.

Algorithm 5.10 Process dataset for destination cleaning

```
1: i ← 1
2: Add column destination_temp to dataset
3: while i < number of rows in dataset do
4:     Read ith observation
5:     destination_temp ← Clean(destination)          # Algorithm 5.11
6:     while Same destination, voyage and vessel do
7:         Change destination_temp
8:     end while
9:     if New destination at first moored observation then
10:        Set destination_temp to destination_temp of previous observation
11:    end if
12:    Increment i to first unchanged observation
13: end while
```

To clean the notation of destinations certain strings or characters in the destination field are identified. If these strings or characters occur they are removed. First a special case is changed from ”RTM NL” to ”NLRTM” (the port code for Rotterdam). The first check is for notations that first state the port of origin followed by the port of destination. The port of origin and the following special characters are removed. Vessels use comma’s or forward slashes to indicate special locations or countries. Since only general ports are of interest, these notations are also removed. After this all digits and punctuation is removed from the destinations. Special statuses like moored, for order , VIA NOK etc. are removed. When all these special cases are removed extra spaces are removed and whitespaces at the start and end of the string are trimmed. If a port code with a space is used the space is removed. And when after standardizing the notation an empty string is the result, the value is set to NA. A brief pseudocode is shown in Algorithm 5.11 and some examples in Table 5.7.

Algorithm 5.11 Clean destinations

```
Input: Temp
1: if Pattern in Temp then
2:     Remove pattern from Temp
3: end if                                  # Repeat this for multiple patterns
4: Remove trailing and leading whitespace from Temp
5: if Space in Port Code then
6:     Remove space from Port Code
7: end if
8: if #char in temp = 0 then
9:     Temp ← NA
10: end if
Output: Temp
```

After the notation of the destinations is cleaned, typos are removed and Port Codes are transformed into Port Names. The dataset is looped over and when status moored is encountered the error variable min is set to 1.000.000.000. This variable is used to keep track of the best possible destination. The destination is first passed to the function that compares the destination to names of the ports. This function takes the observation, the first letter of the destination and the variable min as input. It is also possible to pass a temporary destination, which is discussed in a bit. The look up function takes the list of port names that start with the first letter of the destination. It uses stringdistance based on the ”optimal string alignment” method. This method based on edit distance allows for transposition and is therefore best suited to resolve typos[7, 48, 77, 78]. All the entries in the list are compared to the listed destination and the best possible ports are returned. If an exact match is found the function sets the name as the standardized port for the observation and sets a flag to true. The observation is returned to the standardizing function. If not an exact match is found all possible solutions and the minimal error are returned and the standardizing function call the function that checks based on port codes. This function does the same as the port names functions but uses the port codes instead of names. When this still does not result in an exact match a check is performed to check if only one solution is returned. If so this solution is set as the standardized port. When multiple possible solutions are available, A check is performed if substrings of the destination, if it consists out of multiple words, will lead to better results by first removing the last word from the string one by one, followed by removing the first word from the string. These substrings are passed as temporary destinations. When a substring leads to one possible port, this result is used as the standardized destination. When an attempt is made to standardize the name the results are assigned to the previous observations with the same voyage ID and IMO number while the vessel is sailing and to the following observations while the vessel is moored. If an inconclusive result is found, the WPI port database with location information is used to identify a port based on the location of the vessel. Because this database contains a lot of small ports that are enclosed by big ports, for instance the port of Maassluis that is enclosed by the Port of Rotterdam, this does not always returns good results. But it can be used to get an idea where the vessel is located and this can be used in manually deciding on the destination of a vessel. The pseudocode for standardizing is shown in Algorithm 5.12 and some examples in Table 5.7. How to look up names or codes is shown in Algorithm 5.13.

Table 5.7: Examples of cleaned and standardized destinations

Destination	Cleanded destination	Standardized destination
Rotterdam via NOK	Rotterdam	Rotterdam
NLRTM	NLRTM	Rotterdam
RTM, Netherlands	RTM	Rotterdam
Rottesdam	Rottesdam	Rotterdam
Rotterdam Euromax	Rotterdam Euromax	Rotterdam
Rotterdam Anchorage	Rotterdam	Rotterdam
NL RTM	NLRTM	Rotterdam
APM	APM	Maassluis

Algorithm 5.12 Standardize destinations

Input: ERP_AIS_relevant, ports, WPI

```
Add columns to dataset: destination_standard, destination_name, destination_flag
i ← 1
while i ≤ number of observations in ERP_AIS_relevant do
    Read ith observation
    if Vessel is moored and has a cleaned destination then
        Initialize error value
        Find matches based on Port name
        if No exact match then
            Find matches based on Port Code
        end if
        if One possible but not exact match then
            destination_standard ← destination_name
            destination_flag ← true
        else if Multiple words in destination then
            temp ← destination_temp
            while Multiple words in temp do
                Remove last word
                Find matches based on name
                if No exact match then
                    Find matches based on Port Code
                end if
                if One possible but not exact match then
                    destination_standard ← destination_name
                    destination_flag ← true
                end if
            end while
        end if
        if Multiple words in destination and no match then
            temp ← destination_temp
            while Multiple words in temp do
                Remove first word
                Find matches based on name
                if No exact match then
                    Find matches based on Port Code
                end if
                if One possible but not exact match then
                    destination_standard ← destination_name
                    destination_flag ← true
                end if
            end while
        end if
        while Vessel is sailing to current position or moored at current position do
            if Match found then
                Set destination_standard and destination_flag
            else
                Set destination_name
            end if
        end while
    else
        Increment i by 1
    end if
end while
Create column destination_knn
Create training set with labels from WPI and test set of observations with status moored from ERP_AIS_relevant
Find destination based on location
i ← 1
while i ≤ number of observations in ERP_AIS_relevant do
    Read ith observation
    if Observation in test set and multiple matches base on name and code then
        Set destination_knn
        while Traveling to current location do
            Set destination_knn
        end while
    end if
    Increment i by 1
end while
```

Output: ERP_AIS_relevant

Algorithm 5.13 Look up port names or codes

Input: Observation from ERP_AIS_relevant, first letter of destination, list of port names or port codes, temp

```
ports ← Retrieve ports with same first letter in name or port code
for Every port in ports do
    if temp = NA then
        error ← stringdistance between port and ERP_AIS_relevant$destination_temp
    else
        error ← stringdistance between port and temp
    end if
    if error < min then
        if error = 0 then
            ERP_AIS_relevant$destination_standard ← port
            ERP_AIS_relevant$destination_flag ← true
            ERP_AIS_relevant$destination_name ← NA
            break
        else
            min ← error
            ERP_AIS_relevant$destination_name ← port
        end if
    else if error = min then
        Add port as possibility to ERP_AIS_relevant$destination_name
    end if
end for
```

Output: Observation of ERP_AIS_relevant, min

5.6.8. Shipping lines

After this function most names have been standardized however some destinations still are not standardized because for instance the name of a terminal is stated as destination, like APM in the Port of Rotterdam. Therefore some manual processing is needed which is discussed at the end of this section. First the attribute shipping lines is added to the dataset. Because shipping lines invest in terminals[63], the shipping line of a vessel may have an influence on waiting times. Because terminals may give vessels of their own shipping line priority over vessels of other shipping lines. Most shipping lines state an indication of their name in the name of a vessel. Therefore the name of a vessel is checked for substrings stating the name of a shipping line. If a certain substring is present the shipping line is set to the one related to that substring. If no possible shipping lines are found the shipping line is set to ”UNKOWN”. An pseudocode is shown in Algorithm 5.14. Shipping lines AL, APL, ASTRO, ATLANTIC, BBC, BF, BG, BOMAR, BOX, CAP SAN, CMA CGM, CONMAR, CONTI, COSCO, CSCL, EXPRESS, DS, E.R., ECL, EM, EVERGREEN, FRISIA, BRIDGE, HANJIN, HANSA, HS, HYUNDAI, ICE, JORK, JPO, SCHEPERS, MAERSK, MAX, MOL, MSC, MV, NORDIC, NYK, OOCL, OPDR, PHOENIX, THALASSA, WES, WILSON, XIN, YM and ZIM are checked for. These shipping lines are retrieved by looking at the names of all the vessels in the dataset and identifying recurring patterns. An outline for the algorithm can be found in Algorithm 5.14 and some examples in Table 5.8.

Table 5.8: Examples of shipping lines

Vessel name	Shipping line
MSC EYRA	MSC
Pirita	Unkown
Atlantic Comet	Atlantic
Maersk Missouri	Maersk
Ever unity	Evergreen

Algorithm 5.14 Shipping lines

Input: ERP_AIS_relevant

```
Create column shipping_line with value ”Unkown”
i ← 1
while i < number of rows in ERP_AIS_relevant do
    Read ith observation
    if pattern1 in vessel name then
        shipping_line ← pattern
        while Same vessel do
            shipping_line ← pattern
        end while
    else if patternx in vessel name then          # Repeat for multiple patterns
        shipping_line ← patternx
        while Same vessel do
            shipping_line ← patternx
        end while
    else
        Find next vessel
    end if
end while
```

Output: ERP_AIS_relevant

5.6.9. Manual check and manipulation

Unfortunately the above data manipulation is not able to standardize all destinations and also some voyages are not removed by the dataset while they are not relevant. For instance when a vessels lays at an anchorage waiting for orders at the Port of Rotterdam and then goes into port to pick up containers this is seen as a voyage. Therefore a manual check of the data is performed. Some of these problems can be resolved when using the method multiple times as is shown in Chapter 9. Unfortunately after the first run some manual processing needs to be done and these are explained. These steps are done thoroughly and also repeated twice.

First a data frame is created that contains every combination of IMO number, voyage ID and the destinations, either standardized or not. The routes in the data frame are checked. Every route should start with a standardized destination that is not Rotterdam and finish in a standardized destination that is the port of Rotterdam. If the route starts with Rotterdam the captain forgot to change the location (also possible with other destinations but at the first run this can not be checked). If routes are very short, max 2 destinations, that specific route is checked in the AIS dataset. Vessels may be laying at anchor for almost the entire route and sail into a port to pick up new containers for example. Furthermore a check for wrongly standardized destinations is performed. An excel file named ”dest_rename.xls” is created and the columns are named: the first column ”IMO”, the second ”ROUTE”, the third ”Wrong”, the fourth ”Right”, the fifth ”Latitude” and the sixth ”Longitude”. The IMO and Route ID are put in the corresponding column if a change is needed. The wrong destination is put in column ”Wrong” and the correct destination in ”Right”. When the captain forgets to change a destination, this needs to be done manually for the first observation where the vessel is moored in the port. The latitude and longitude of that observation are set in the corresponding columns. When a specific route needs to be removed, the IMO and route are set in the corresponding columns and ”REMOVE” is put in column ”WRONG”. When all the observations of a vessel needs to be removed it is not necessary to add a row for every voyage. The ROUTE column is left blank. Changes are added in the order they are encountered. This makes the processing easier. When the excel file is finished, this file is read into R as a data frame. The resulting data frame is shown while the next function runs, to check if every change is performed. An example of the resulting excel file is shown in Appendix D

In Algorithm 5.15 an algorithm is shown to process the excel file. The first function first runs through the data frame loaded from the excel file to identify which voyages or vessels need to be removed. It creates a data frame of vessels with a corresponding voyage ID that needs to be removed. When no voyage ID is set all the observations of that vessel needs to be removed. When the data frame is created the AIS dataset is looped over, setting the IMO of observations that need to be removed to NA and after looping over the dataset all observations with an IMO set to NA are removed. After removing these observations, the AIS dataset is looped over again. If the destination stated in a observation is present in the data frame from the excel file this data frame is looped over to search for the corresponding entry. The destination is changed accordingly. When the entire AIS dataset is processed, dataset is again fed to the function that standardizes the destinations to remove possible typos made in the excel file.

Algorithm 5.15 Process manual changes

Input: ERP_AIS_relevant, manual_rename
Create data frame remove
for all Rows in manual_rename **do**
  Read row
  if Remove is **true then**
    add IMO and route ID to remove
  end if
end for
for all Rows in remove **do**
  Find matching observations and set IMO to NA
end for
Remove observations with IMO = NA
for all Observations in ERP_AIS_relevant **do**
  if Observation needs to be changed **then**
    Change observation
  end if
end for
ERP_AIS_relevant ← Standardize destination *# Algorithm 5.12*
Output: ERP_AIS_relevant

After all these steps the AIS dataset is cleaned, manipulated and all the attributes of AIS except draught that are error prone have been checked and if needed changed. To check if any other manipulations are needed another summary of the dataset is created. The output is shown in Output 5.4

Output 5.4: Output of the summary function. For every pre-processed variable the minimal and maximum values are shown. Furthermore it shows the mean, median and values at the 1st and 3rd quartile second line the number of variables.

```
   timestamp                      mmsi                status
   Min.   :2014-08-25 21:01:00   Min.   :209177000   Min.   : 0.000
   1st Qu.:2015-01-30 12:15:45   1st Qu.:218572000   1st Qu.: 0.000
   Median :2015-05-28 16:06:30   Median :255805578   Median : 0.000
   Mean   :2015-05-02 11:13:06   Mean   :324278317   Mean   : 1.881
   3rd Qu.:2015-08-05 23:02:00   3rd Qu.:354776000   3rd Qu.: 5.000
   Max.   :2015-10-31 14:21:00   Max.   :636092637   Max.   :15.000

      speed            latitude        longitude          course
   Min.   :  0.000   Min.   :17.76   Min.   :-10.344   Min.   :  0.0
   1st Qu.:  0.000   1st Qu.:51.68   1st Qu.: -1.151   1st Qu.: 80.0
   Median :  8.300   Median :53.35   Median :  3.778   Median :197.0
   Mean   :  7.474   Mean   :52.99   Mean   :  5.136   Mean   :182.2
   3rd Qu.: 14.500   3rd Qu.:54.60   3rd Qu.:  9.640   3rd Qu.:270.0
   Max.   :186.000   Max.   :63.65   Max.   : 40.309   Max.   :359.0
```

```
      heading       destination           draught
   Min.   : -1.0   Length:205884      Min.   :  1.00
   1st Qu.: 85.0   Class :character   1st Qu.: 66.00
   Median :205.0   Mode  :character   Median : 75.00
   Mean   :188.6                      Mean   : 80.78
   3rd Qu.:275.0                      3rd Qu.: 91.00
   Max.   :511.0                      Max.   :244.00
   NAs    :5

        eta                           imo             name
   Min.   :2014-06-14 11:00:00   Min.   :8201648   Length:205884
   1st Qu.:2015-01-30 07:00:00   1st Qu.:9287704   Class :character
   Median :2015-05-28 19:00:00   Median :9355422   Mode  :character
   Mean   :2015-05-02 05:04:27   Mean   :9363402
   3rd Qu.:2015-08-05 16:00:00   3rd Qu.:9454242
   Max.   :2015-12-15 20:01:00   Max.   :9708851

     callsign             type             ais_type          bow
   Length:205884      Length:205884      Min.   : 0.00   Min.   :  0.0
   Class :character   Class :character   1st Qu.:71.00   1st Qu.:117.0
   Mode  :character   Mode  :character   Median :71.00   Median :134.0
                                         Mean   :72.33   Mean   :126.6
                                         3rd Qu.:74.00   3rd Qu.:146.0
                                         Max.   :79.00   Max.   :273.0

      stern             port          starboard
   Min.   :  0.00   Min.   : 0.00   Min.   : 0.00
   1st Qu.: 10.00   1st Qu.:11.00   1st Qu.:10.00
   Median : 14.00   Median :12.00   Median :11.00
   Mean   : 63.78   Mean   :14.56   Mean   :13.79
   3rd Qu.:121.00   3rd Qu.:17.00   3rd Qu.:16.00
   Max.   :341.00   Max.   :41.00   Max.   :40.00

        ATA                          route_id      est_traveltime
   Min.   :2014-08-31 09:17:00   Min.   :  1.00   Min.   :-121.17
   1st Qu.:2015-02-04 01:00:00   1st Qu.: 10.00   1st Qu.:  40.97
   Median :2015-06-02 18:16:00   Median : 50.00   Median :  89.98
   Mean   :2015-05-06 22:55:16   Mean   : 69.56   Mean   : 110.61
   3rd Qu.:2015-08-11 02:01:00   3rd Qu.:110.00   3rd Qu.: 150.98
   Max.   :2015-10-31 14:21:00   Max.   :356.00   Max.   : 998.00

    avg_speed         obs_speed        d_speed              length
   Min.   : 0.0000   Min.   : 1.00   Min.   :-167.40000   Min.   :100.0
   1st Qu.: 0.6917   1st Qu.:12.00   1st Qu.:  -0.40000   1st Qu.:138.0
   Median : 8.2500   Median :12.00   Median :   0.00000   Median :149.0
   Mean   : 8.1475   Mean   :11.66   Mean   :  -0.08636   Mean   :190.5
   3rd Qu.:14.1500   3rd Qu.:12.00   3rd Qu.:   0.30000   3rd Qu.:194.0
   Max.   :54.4000   Max.   :12.00   Max.   : 167.70000   Max.   :400.0
                                                          NAs    :125
```

```
      width          traveltime     destination_temp
 Min.   :18.00   Min.   :  0.00   Length:205884
 1st Qu.:22.00   1st Qu.: 44.22   Class :character
 Median :23.00   Median : 92.28   Mode  :character
 Mean   :28.37   Mean   :107.70
 3rd Qu.:28.00   3rd Qu.:151.32
 Max.   :60.00   Max.   :336.00
 NAs    :125

 destination_standard   destination_name   destination_flag
 Length:205884          Length:205884      Mode :logical
 Class :character       Class :character   FALSE:589
 Mode  :character       Mode  :character   TRUE :205295
                                           NAs  :0
 destination_knn    shipping_line
 Length:205884      Length:205884
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Still some NA's remain in the dataset. Specifically for heading and the dimensions of the vessel. Also in the speed variables a peculiar occurrence is observed. A vessel has travelled at a speed of 186 km/h which is impossible. Also the max of AIS status is 15, which is the default status. Since speed and heading will be removed in the dimension reduction (see next section) no further action is needed. However a closer look is taken at the AIS status and dimensions. For the AIS status message it turns out that this status is used in 6 observations for a vessel when moored in Hamburg so the status of these observations is changed to 5. The NA's for width and length all belong to a single vessel, the MSC Maureen with IMO number 9251717. According to the database of marinetraffic this vessel is 299.9 m in length and its width is 40m[3]. These values are set manually.

[3] https://www.marinetraffic.com/en/ais/details/ships/shipid:416228/imo:9251717/mmsi:355216000/vessel:MSC%20MAUREEN

5.7. Dimension reduction

One tends to believe that adding more features will improve the fitted model, however this only happens when the added features are truly associated with the response. When features that are not truly associated with the response are added, the fitted model will deteriorate. So called noise features increase the dimensionality which increases the chance of overfitting. This is called the ”curse of dimensionality”[29]. Therefore the input variables are looked at again. The following attributes are identified as possible input variables.

- Latitude
- Longitude
- Current speed
- Change in speed over the last 3 hours
- Average speed based on last 12 hours
- Observations used for calculating the average speed
- Length and breadth of the vessel
- ETA provided by the vessel
- IMO number
- Shipping line
- Draught
- Navigational status

As already mentioned the ETA is a date and thus is not useful to use as an input since its value can not be generalized. So the ETA is replaced with the estimated travel time of the vessel. The speed variables are removed because this variable varies to much. The speed could change in seconds and therefore does not have a lot of predictive value. The IMO is also removed because this is a unique identifier and therefore makes the model less generic. So the resulting input variables are:

- Latitude
- Longitude
- Draught
- Dimensions of the vessel
- Estimated travel time
- Navigational status
- Shipping line

5.8. Resulting dataset

After executing the steps, as mentioned in the previous section, the resulting dataset is clean and ready to be used for route identification. The dataset started as a set with 5,95 million observations. After removing vessels from the dataset that have not visited the Port of Rotterdam or are not a container vessel 5,26 million observations remained. Because the ETA for a vessel remained the same over multiple voyages the voyages with an inaccurate ETA are removed. By doing this the dataset is reduced to 3,28 million observations. Because only a short time horizon is needed for the predictions, all observations with a travel time that exceeded 14 days are removed, now the dataset contains 1 million observations. After removing incomplete and short routes just 482.284 observations remained. Among these observations are also a lot of voyages with gaps in the observations, so also these voyages are removed. This leads to another large reduction, leading to a dataset of 216.424 observations. After the manual removal of routes that are at anchor for the most of the time or with destinations that did not change, the dataset that is used only consists out of 202.328 observations. So only 3,4% of the original dataset can be used for route prediction.

5.9. Conclusion

In this chapter the first two sub research questions are answered: *What are possible issues with the data quality of AIS messages?* and *How can AIS messages be pre-processed to improve the data quality so it can be used for route identification and ETA prediction?*. The findings are discussed in this section.

In Section 5.2 possible issues that hamper the quality of AIS data are discussed. This discussion is based on the literature review. In the conclusion the findings from this discussion are combined with the findings from processing the data. Multiple issues are identified that affect the data quality. First the unique identifiers as transmitted by the vessels may be wrong, especially the MMSI number was very error prone in our dataset. So for vessel identification it is important to use the identifier that is least error prone. Depending on the research it might be necessary to filter the dataset based on vessel types. However these types are not always inputted correctly, the class might be too general, too specific, wrong or absent. So it is necessary to cross reference the vessel types. The database of the International Maritime Organization[4] is used. Other features of the dataset that might have issues are the features related to the dimensions of the vessel. These variables may be used to calculate the dimension of the vessel but they might display false information, no information or wrong correlation between the variables. From the literature review it became evident that vessels sometimes broadcast positional information that is outside the possible range. In the AIS dataset this was not a problem but it needs to be checked. The navigational status of a vessel needs to be inputted manually and is not checked by the AIS system aboard a vessel. Therefore the navigational status is very error prone and needs to be checked by the researchers based on other variables in an AIS message. Another variable that is manually inputted is the draught of the vessel, but just like the navigational status this variable is very error prone. No draught or a draught of 0 meters is broadcasted. For these errors checks are performed in the dataset. However it might also be possible that the draught is off by several meters, but this can not be checked because a historical dataset is used. The most error prone variable is the destination of a vessel. Since the only limitation on this variable is the amount of characters available, the destination is stated in all different kind of forms. As a number or abbreviated for instance. The displayed information may also be fake, not inputted or not updated. Because of this unstructured nature of the data a lot of effort needs to be put in standardizing the destinations during the data processing. The last variable that has a negative effect on the quality of an AIS dataset is the ETA. The ETA is also communicated manually (although in the dataset it is unclear if it is communicated by the vessel or the shippers agent), so also ETA's may not be updated causing multiple routes to have the same ETA, as observed while processing, or ETA's that are in the past for a different port visit. Another possibility are ETA's in the very distant future.

The last issue with AIS data lies in the completeness of the data. During processing a lot of voyages were incomplete or had large gaps. In the literature review researches stated that AIS is broadcasted on regular short intervals and that due to the improvements in the network the maximal latency was about a minute so coverage should not be an issue. Since the data was collected, decoded and sampled by another party, it is unclear whether the problem is related to AIS messages in general and literature is wrong or to this dataset due to errors in the decoding and sampling process of the other party. The fact remains that data can be incomplete. In the ideal case the missing messages are generated by interpolation for example, however due to time constraints incomplete voyages are deleted.

[4] https://gisis.imo.org/Public/SHIPS/Default.aspx

As becomes evident in this chapter a lot of data from AIS messages is of a low quality and needs to be processed to improve the data quality. Especially variables that need to be inputted manually, such as the status and destination, are very error prone and therefore need to be manipulated. By following the workflow as presented in Figure 5.11, one can get an AIS dataset with such a quality that it can be used for route identification and ETA prediction. However depending on the dataset additional steps might be necessary or some steps can be omitted. A discussion of the steps is presented in Sections 5.4, 5.5 and 5.6. An implementation can be found in Appendix C. Some manual steps are needed to increase the data quality, so the cleaning and manipulation algorithms can be improved. These improvements are discussed in Chapter 9.

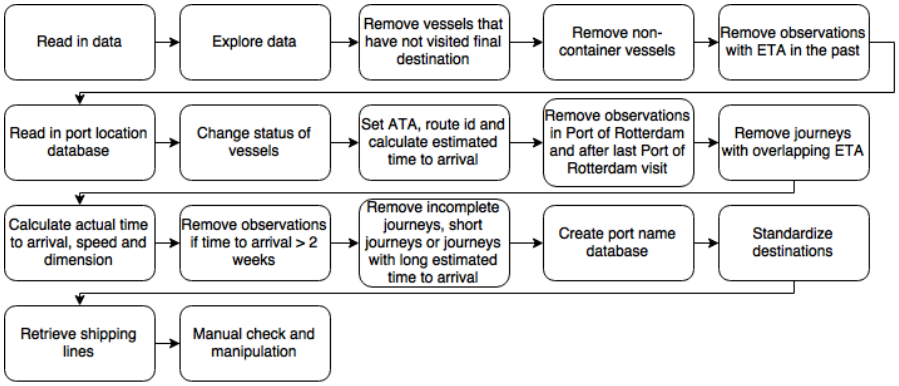


Figure 5.11: Manipulation steps of framework to improve the quality of an AIS dataset

6

Route identification

In their research Dobrkovic et al. concluded that in order to improve the predictions regarding container vessels 4 areas needs to be improved, among which the discovery and inclusion of behavioral patterns[12]. According to Lane et al. ports are visited in a particular order by container vessels[35], so this constitutes some kind of behavior. Therefore routes are included into the predictions. In this chapter we answer the question "How can a set of possible routes of a container vessel be identified using pre-processed AIS data?". First two route prediction methods are assessed and the insights of these methods are used to develop a framework to identify a set of possible routes for a vessel based on pre-processed AIS messages. This framework is presented in Figure 6.1.

6.1. Traffic Route Extraction and Anomaly Detection

Pallotta et al. have developed a methodology that makes it possible to characterize maritime traffic with a unsupervised learning strategy based on AIS data. This method is able to extract routes. These route can be used to gather useful information on daily patterns and travel times. But also to associate historical voyage patterns to vessels. This information can also be used to predict the route of a vessel at any given time based on the behavior of other vessels in the past on the same route[57].

In the research field of maritime anomaly detection vectorial representations of traffics are more and more adopted. With this representation trajectories are considered as a straight path between two waypoints. This allows for vessel motions to be represented compact on a global scale[57].

The method as developed by Pallotta et al. is called "Traffic Route Extraction and Anomaly Detection"(TREAD), this method learns unsupervised and automatically a statistical model for traffic in the maritime domain based on AIS data. The knowledge of the traffic is created and updated by the sequential input of AIS messages and stored in vessel objects. For this methodology a bounding box is selected that corresponds to an area that is under surveillance. The AIS observations are used to form waypoints with the use of clustering. These waypoints can be a stationary points or entry/exit points, where the vessel enters or exits the area under surveillance. Stationary points can be ports, anchorages or other off shore points of interest. These waypoints are then linked and thus leads to the detection of routes[57].

When a vessel enters the area that is under surveillance, this vessel is detected by TREAD. If the vessel is already in the list of vessel objects the vessel object is updated, otherwise the vessel is added to the list. Every observation of a vessel is used to check and update the waypoints if needed. Then the waypoints are used to check if new routes need to be added to the route objects. Routes can be between two stationary points, between a stationary point and a entry/exit point or between two entry/exit points. The route objects do not only keep track of which vessels have travelled over that route but also keeps track of static and movement features of the vessels that used the route[57].

TREAD is able to predict the future route of a vessel or to detect anomalous behavior. With route classification Pallotta et al. assign a probability to every route that can be travelled from the current position of the vessel. This classification is done based on the partial route that has already been observed by TREAD. Using a Bayesian approach the probability for each route is calculated[57].

The methodology of Pallotta et al. have produced some useful insight in how to do route classification. However their method is deemed overly complex for the goal of the research. Pallotta et al. also cluster turning points of vessels and a lot of other intermediate waypoints. The clustering of al those extra points is needed for the anomaly detection but also makes the method very complex. Since only port visits are of interest, other methods are assessed to see if these are more suitable.

6.2. Hidden Markov Model

Vessels visit certain ports in a certain sequence over time. These sequences can be characterized using a Hidden Markov Model. These models can be applied on the individual level, so one model for each vessel. A Markov model represents a discrete-time stochastic process where the probability distribution over states at a particular time step depends only on the state at the immediately preceding step[35]. So in our case the probability that a port is visited next is based on the last visited or current port. These probabilities are captured in a transition probability matrix. The probabilities are estimated by taking the proportion of times each transmission is made in the training set. But because a lot of ports exist around the world and the amount of training data is limited, the accuracy of the probabilities in the transition probability matrix will also be limited. Using a Bayesian approach this problem can be improved upon. The distribution on each row of the matrix is updated with Bayes' theorem using a prior distribution and the observed data[35].

Although the Hidden Markov Model is a very good approach to predict next destinations of a vessel, it is also computationally very expensive and fails to utilize the full extent of an AIS message. The location information is used but not the destination that is stated in the AIS message. So the insights from this model that routes can be represented sequential and that these routes are visited multiple times over a timespan are used. Furthermore it shows that models for prediction can be created on different levels. But since it is computationally expensive and does not utilize the destination as stated by the vessel, a simpler and less computationally expensive method is developed.

6.3. Route identification framework

Both TREAD and the Hidden Markov Model provide a complex and computationally expensive method to construct and classify or predict routes. Because these methods are too complex for the research problem a simpler method is introduced that is also computationally less expensive, based on AIS messages. A framework for this methodology is presented in Figure 6.1. The method requires an AIS dataset that is used to create a route database of all the different routes. In this route database is stored which unique routes exist, which vessels have travelled which route and how many times a route is travelled. This route database is then used to identify a set of possible routes for a vessel. Each step is discussed in more detail in the following sections. Subsection 6.3.1 discusses the first step, Subsection 6.3.2 discusses the second step and Subsection 6.3.3 discusses the remaining steps. A coding example of this methodology is presented in Appendix C on lines 1562 through 1717. A running example is provided in Subsection 6.3.3.

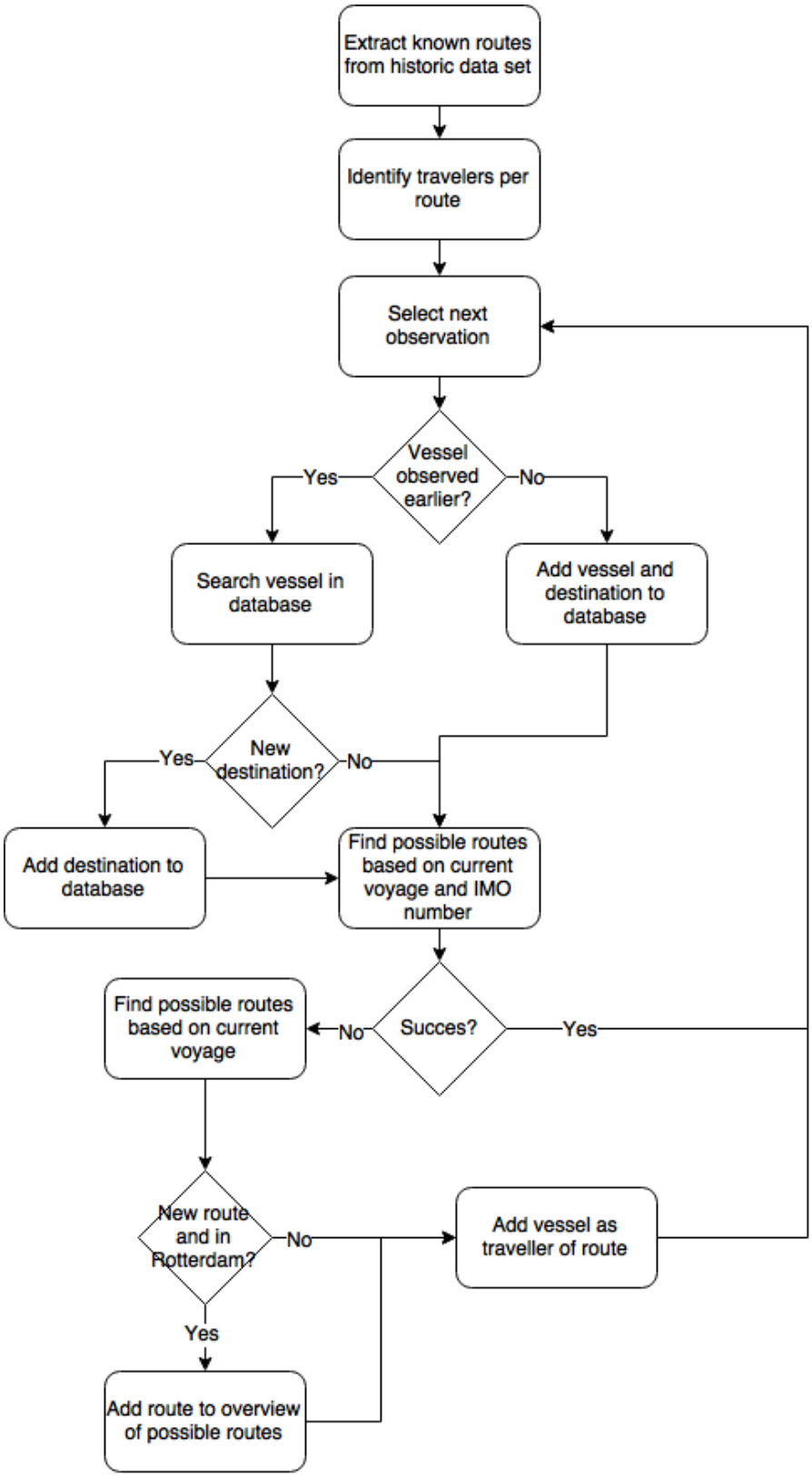


Figure 6.1: Route identification framework

6.3.1. Creating routes

For the methodology a historical AIS dataset is needed in which the destinations are represented in a standardized way, every stop has at least one observation where the status is set to moored and every voyage per vessel has an unique identifier. In Section 5.5 and Section 5.6 is discussed how to obtain such a dataset. The method creates a data frame that contains the route for every voyage of every vessel. The method loops over the historical dataset and adds a row to the data frame for every voyage of every vessel. When the method encounters a new voyage for a new vessel the method stores the IMO number of the vessel and the voyage ID in the data frame. The method adds the first stated standardized destination to the route column. Every standardized destination that is stated during the voyage is pasted at the end of the string. Now the method has created a sequential representation of the ports visited during the voyage. When the function finishes a data frame that contains this sequential representation for every voyage of every vessel is available. Algorithm 6.1 shows the pseudocode for the first part of the method.

Algorithm 6.1 Create routes

Input: AIS Database with standardized destinations

```
Create empty data frame with columns: IMO, route_id and route
for Every voyage of a vessel do
    Create a row in data frame with IMO number of vessel and route_id of voyage
    Identify and store route in created row
end for
```

Output: Data frame containing the route of every voyage

6.3.2. Identifying travelers of routes

The data frame that is created with the previous function is used as an input to the next function that creates the route database, an overview of how many times a route is travelled and by which vessels. A list is created that contains a data frame with all the unique routes and the variable times taken set to zero and it contains another list in which for every route the vessels are stored that have travelled that route. Every component of the list with vessels is named after a route. The method then loops over the created data frame and for every route loops over the data frame created in the previous function. Every time it encounters the route we are searching for the counter is incremented by one. The method also checks if the corresponding IMO number is stored in the list of vessels that have travelled that route, if it has not. When the loops are finished the data frame is order descending based on the number of times a route is travelled. A pseudocode is shown in Algorithm 6.2.

Algorithm 6.2 Identify unique vessels per route and count times a route is travelled

Input: Data frame containing the route of every voyage

```
Create data frame containing unique the routes and set times taken for every route to 0
Create list with n entries                          # n is number of unique routes
Name entries of list after routes
for Every unique route do
    for Every voyage do
        Read voyage
        if Voyage == Route then
            Increment times taken of route by one
            if New vessel then
                Add IMO to entry of route in list
            end if
        end if
    end for
end for
```

Output: Route database containing routes, list of routes with IMO of vessel that traveled routes

6.3.3. Route identification

Now the route database is created, real time AIS data is processed. A proof of concept is shown by passing the observations of the top six most visited routes to the identification function. A data frame is created that contains AIS messages from these route and the data frame is ordered based on the time stamp so the data frame mimics real time data. To keep track of all the vessels in a defined area of observation a vessel database is created that contains all the vessels that are in the area under observation. When an AIS message is received from a vessel, the presence of that vessel in the vessel database is checked. If the vessel is moored in the Port of Rotterdam or still communicating the same destination, nothing happens. When the stated destination is different from the last known destination, this new destination is added to the sequential representation. If the vessel is not in the data frame a row is added to the vessel database containing the IMO number of the vessel and its current destination. This sequential representation is compared to the first n destinations of every route, based on the amount of destinations that are in this sequential representation. For every identified route is checked if the vessel has travelled the route. By this comparison and check a set of possible routes is printed based on the current voyage of the vessel and its IMO number. When no possible routes are found based on the current voyage and IMO number, the check of the IMO number is ignored and a set of possible routes based on the current voyage is produced and at the end of the voyage the IMO number is added to the relevant list in the route database. If still an empty set is produced the vessel is traveling a new unobserved route. When a vessel is moored in Rotterdam en thus finished the voyage, the vessel is removed from the vessel database. By comparing the current voyage to the relevant number of first destination of every route, more routes are excluded when a vessel is further along his route. Thus reducing the set of possible routes until one route remains.

Algorithm 6.3 Route identification

Input: Route database containing routes and times taken, list of routes with IMO of vessel that traveled routes, AIS data

```
Create empty vessel database with columns IMO and route
for Every AIS message do
    Read AIS message
    if Vessel is under observation then
        if New destination stated then
            Add destination to route
        end if
    else
        Add row to vessel database and set IMO number and route as the destination
    end if
    Identify possible routes based on IMO number and current voyage
    if No routes possible based on IMO number and current voyage then
        Identify possible routes based on current voyage
    end if
    if Vessel is moored in final destination then
        if Vessel has not traveled identified route before then
            Add IMO number of vessel to entry in list of routes
        end if
        Remove row containing vessel and route from vessel database
    end if
    print Set of possible routes and identification criteria
end for
```

Running example

In this section a simplified running example of the route identification algorithm is shown. The Table 6.1 shows a simplified route database containing only two routes, as could be created by the first three steps of the framework as presented in Figure 6.1. The following table is the vessel database where the vessel under observation are stored. This is followed by a table containing a simplified AIS message. Following this message are tables containing the updated vessel database and tables containing the result of the identification and the next AIS message.

Table 6.1: Route Database

Route	IMO Numbers
Dublin → Rotterdam	1
Dublin → Southampton → Rotterdam	1, 2

Table 6.2: Vessel database

Imo number	Voyage

Table 6.3: first AIS message

Imo number	Destination	Set of possible routes
1	Dublin	

Table 6.4: Vessel database

Imo number	Voyage
1	Dublin

Table 6.5: Result & next AIS message

Imo number	Destination	Set of possible routes
1	Dublin	Dublin → Rotterdam, Dublin → Southampton → Rotterdam
2	Dublin	

Table 6.6: Vessel database

Imo number	Voyage
1	Dublin
2	Dublin

Table 6.7: Result & next AIS message

Imo number	Destination	Set of possible routes
2	Dublin	Dublin → Southampton → Rotterdam
1	Southampton	

Table 6.8: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin

Table 6.9: Result & next AIS message

Imo number	Destination	Set of possible routes
1	Southampton	Dublin → Southampton → Rotterdam
3	Dublin	

Table 6.10: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin
3	Dublin

Table 6.11: Result & next AIS message

Imo number	Destination	Set of possible routes
3	Dublin	Dublin → Rotterdam, Dublin → Southampton → Rotterdam
2	Rotterdam	

Table 6.12: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin → Rotterdam
3	Dublin

Table 6.13: Result & next AIS message

Imo number	Destination	Set of possible routes
2	Rotterdam	Dublin → Rotterdam
1	Rotterdam	

Table 6.14: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin

Table 6.15: Result & next AIS message

Imo number	Destination	Set of possible routes
1	Rotterdam	Dublin → Southampton → Rotterdam
3	Dublin	

Table 6.16: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin

Table 6.17: Result & Last AIS message

Imo number	Destination	Set of possible routes
3	Dublin	Dublin → Rotterdam, Dublin → Southampton → Rotterdam
3	Rotterdam	

Table 6.18: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin → Rotterdam

Table 6.19: Result

Imo number	Destination	Set of possible routes
3	Rotterdam	Dublin → Rotterdam

6.3.4. Performance assessment & results

To show the full functionality of the method a list of routes is created without the IMO numbers stored. This makes it possible to show that identifications are made both based on current voyage of a vessel and based on routes a vessel has taken in the past and the current voyage. The empty list also makes it possible to asses the accuracy of the algorithm.

The methodology can give a wrong set of possible routes in two cases. The first case is when a vessel travels via a route that the vessel has not travelled before, but has already travelled other routes that are similar at the start. In this case, due to the assumption that vessels are more likely to take the same route, a possible set of routes is identified based on the IMO number of the vessel and the current voyage. The correct route is not identified in this set since its IMO number is not that route. In this case at a certain point during the voyage the identification changes from based on IMO number and voyage to based on only voyage and a correct set of possible routes is identified. This characteristic can be used to asses the accuracy.

The second case that produces a wrong set of possible routes is when a new port is build or a vessel travels via a new sequence of ports. In this case vessels travel via a route that has never been observed before. Up to the point the vessel start traveling to the new port or uses an unique sequence, a wrong set of possible routes will be identified, after this point the algorithm is not able to identify a possible set of routes. When the first vessel completes a route with the new port in it, the route should be added to the database.

The accuracy of the algorithm is assessed by counting the times the first case occurs, since a historical dataset is used, new routes do not occur and observing the second case is thus not possible. Dividing the number of changes by the amount of voyages in the data shows that the accuracy is: 99.3%. So the identification algorithm performs very good. For the other 0.7% the algorithm shows a wrong possible set of routes first, but when it becomes clear that the vessel is traveling a route that is not in this wrong possible set of routes, the identification is done based only on the current voyage and the correct route will be in the new possible set of routes.

Some examples of the output the method produces are shown. These examples are used to demonstrate that the method is able to identify routes based on IMO, current voyage and stated destination or only on the current voyage and stated destination. The examples also show that the method is able to identify multiple routes when multiple routes are possible based on the current voyage and stated destination and can handle multiple vessels at the same time.

Output 6.1: Output for six consecutive AIS messages showing the method is able to handle multiple vessels at the same time

```
Set of possible routes for vessel 9461489 during voyage 1 based on
  ↪ current voyage and destination is:
HAMBURG -> ROTTERDAM

Set of possible routes for vessel 9665633 during voyage 4 based on
  ↪ current voyage and destination is:
FELIXSTOWE -> HAMBURG -> ROTTERDAM

Set of possible routes for vessel 9264714 during voyage 50 based on
  ↪ current voyage and destination is:
MOERDIJK -> ROTTERDAM

Set of possible routes for vessel 9461489 during voyage 1 based on
  ↪ current voyage and destination is:
HAMBURG -> ROTTERDAM

Set of possible routes for vessel 9665633 during voyage 4 based on
  ↪ current voyage and destination is:
FELIXSTOWE -> HAMBURG -> ROTTERDAM

Set of possible routes for vessel 9264714 during voyage 50 based on
  ↪ current voyage and destination is:
MOERDIJK -> ROTTERDAM
```

The example in Output 6.1 shows that that multiple vessels can be handled consecutively. The next example in Output 6.2 shows in the first voyage that when multiple routes are possible for a vessel, multiple routes are identified. The example also shows that only routes travelled are taken into account when possible, this is shown by the second voyage in the example where the vessel have previously travelled through the area of observation and thus only this route is predicted. When it becomes evident that a vessel is traveling a different route then identified in previous sets of possible routes based on the IMO number and current voyage, the algorithm changes back to identifications based on current voyage as shown with the third voyage. Since the vessel has now travelled both routes from Dublin, also both routes are identified again. And in the end when more information is known about the route a single route is identified as is shown in the fourth voyage.

Output 6.2: Output of multiple voyages for a single vessel. The first voyage shows multiple routes are identified. Second voyage shows possible routes are excluded when incorporating IMO. The third voyage shows changing back to identification based on current voyage when needed and the last voyage shows excluding routes when more information about the voyage is known.

```
#########First voyage
Set of possible routes for vessel 9287699 during voyage 36 based on
  ↪ current voyage and destination is:
DUBLIN -> ROTTERDAM
DUBLIN -> SOUTHAMPTON -> ROTTERDAM

#########Second voyage
Set of possible routes for vessel 9287699 during voyage 39 based on
  ↪ IMO number, current voyage and destination is:
DUBLIN -> SOUTHAMPTON -> ROTTERDAM

#########Third voyage
Set of possible routes for vessel 9287699 during voyage 48 based on
  ↪ IMO number, current voyage and destination is:
DUBLIN -> SOUTHAMPTON -> ROTTERDAM

Set of possible routes for vessel 9287699 during voyage 48 based on
  ↪ current voyage and destination is:
DUBLIN -> ROTTERDAM

#########Fourth voyage
Set of possible routes for vessel 9287699 during voyage 50 based on
  ↪ IMO number, current voyage and destination is:
DUBLIN -> ROTTERDAM
DUBLIN -> SOUTHAMPTON -> ROTTERDAM

#Next AIS Message

Set of possible routes for vessel 9287699 during voyage 50 based on
  ↪ IMO number, current voyage and destination is:
DUBLIN -> SOUTHAMPTON -> ROTTERDAM
```

6.4. Conclusion

In this chapter the question *How can a set of possible routes of a container vessel be identified using pre-processed AIS data?* is answered and a framework is presented to identify the route of a container vessel with the use of pre-processed AIS data. Insight are taken from the research of Pallotta et al.[57] and Lane et al.[35] and used to create a simpler and computationally less expensive method. With this method a set of possible routes is identified based on pre-processed AIS messages for all vessels that sail to the Port of Rotterdam. The methodology consists of steps as presented in Figure 6.1. First a data frame is created that consists the route of every voyage from a historical dataset that contains pre-processed AIS messages. Then all unique routes are identified and also which route was travelled by which vessels, these are stored in a route database. If this route database is created, pre-processed AIS messages are read and a set of possible routes for a vessel is identified based on its current voyage and possibly IMO number. Using this methodology an accuracy of 99.3% can be achieved, meaning that in 99.3% of the case the correct route is in the set of possible routes. The methodology is not able to identify a set of possible routes for an entire new route, however when such a route occurs the route should be added to the database so it can be used for future identification. In Chapter 7 we show how to use this identification in combination with machine learning to predict the Estimated Time of Arrival of a vessel.

7

ETA prediction

In this chapter the question "How can the ETA of a vessel be predicted with the use of pre-processed AIS data and route identification?" is answered. How to use route identification is discussed first and Algorithm 7.1 is introduced. This framework demonstrates how to do predictions based on routes. The remainder of the chapter is used to select a prediction method, prepare the dataset and discuss the results of the framework.

7.1. Combining route identification and ETA prediction

In Chapter 6 it is demonstrated how a set of possible routes for a vessel can be derived from AIS data. This knowledge is incorporated into Estimated Time of Arrival predictions. This can be one in two wats. Either the route is used as an input variable for the model or the route is used as a selection variable for the model. Both options are discussed and one of the two is selected.

7.1.1. Route as input variable

One of the possibilities is to use the route as an input variable for the prediction model. The route is one of the attributes used to predict the ETA of a vessel in the Port of Rotterdam. With this option the importance of the route for the predictions is unknown and it might also be impossible to capture all the characteristics of the route in a single variable. As already discussed each route has different characteristics such as distance and the amount of ports visited. And for each port also different characteristics exist. All these characteristics influence the travel time of a vessel in a very complex manner. Therefore using a single variable might not capture all characteristics and thus the influence of the route on the predictions might be minimal. Therefore this option does not comply with the objective of the research.

7.1.2. Route as selection variable

The other possibility is to use the set of possible routes as a selection variable. With this option multiple models are created, one for each route that captures characteristics related to the route because only observations from that route are used. The research of Lane et al. have shown that this is a feasible option[35]. The set of possible routes is used to select which models to use for predictions. So all the behavior that is characteristic for that route is captured in the route and it is hypothesized that predictions will be more accurate in comparison to a general model with routes as an input variable. Therefore the routes are used as a selection variable.

In Algorithm 7.1 a framework is shown to do the ETA predictions. For this framework an pre-processed AIS database is needed as produced by the the framework discussed in Chapter 5 and the database that is created with Algorithm 6.2. The AIS database is used to create a training set for each route and a test set that contains observations for all routes. The training sets are used to train a prediction model per route. When the models have been trained, the AIS messages are read from the test set. A message is used to identify a possible set of routes as shown in Algorithm 6.3. This set of possible routes is used to select prediction models to be used for prediction. For every possible route a prediction of the travel time is made with the relevant model. As output the framework gives every prediction made based on the possible set of routes. Due to time limitations the route identification is not improved to route classification. So it is up to the users of the output to make a decision on which prediction to use and to gather any extra information regarding the route to base their decision upon.

Algorithm 7.1 Framework for ETA prediction

Input: AIS database with standardized destinations, database with possible routes and vessels that travelled those routes

1: Create training set for each route and test set from AIS database
2: Create prediction model for each route
3: Read AIS message from test set
4: Identify possible set of routes for vessel *# Algorithm 6.3*
5: Select prediction models based on possible set of routes
6: Do predictions

Output: Prediction for each possible route

7.2. Prediction methods

In this section some prediction methods are discussed that could be used to predict the ETA of a container vessels. Based on this discussion a model is selected in the next section.

7.2.1. Linear regression

Linear regression is the most simple and straightforward approach. Linear regression is based on the assumption that a linear relationship exist between the predictor and a quantitative response. Two types of linear regression exist. Simple linear regression and multiple linear regression. Simple linear regression takes only one predictor and multiple linear regression takes multiple predictors[29]. Since multiple input variables are used in the research, multiple linear regression is more relevant and thus discussed.

Multiple linear regression can be noted as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_p X_p \tag{7.1}$$

Where p stand for the amount of predictors, X_j represent the jth predictor, β_j quantifies the relation between that jth predictor and the response and β_0 is the intercept when $X = 0$. The values for β are unknown and need to be estimated. Based on a training set linear regression finds the regression line that best describes the data points in the training set. This line is found by minimizing the sum of squared residuals. This sum is calculated by taking the square of the error for each datapoint and summing these errors. Linear regression is easy to fit, easy interpretable and does not require a lot of computing power. However one of the downsides of linear regression is the assumption of a linear relationship. However when the relation is not linear the model will fit the data very poorly and no conclusions can be drawn from the predicted values[29, 33].

7.2.2. K-nearest neighbor regression

Another simple approach is K-nearest neighbor(KNN) regression and does not assume a linear relationship. KNN maps the training set on a p-dimensional space, where p is the number of predictors. KNN then maps a point off the test set (x_0) in this space. Based on a given value K, K values that are closest to x_0 are selected. The average of these values is given to x_0. Selecting the value for K decides on the behavior of KNN. When $K = 1$ the results will have a high variance. When $K = p$ KNN behaves as linear regression and is thus biased to a linear relationship. Deciding on an accurate variable for K is related to the bias-variance tradeoff[29, 33].

When choosing the parameters for a prediction model the bias and variance need to be kept as low as possible, however this is not possible. When variance is low the output hardly changes when the input is changed, but when the bias is low the output changes considerably when the input is changed. Therefore a value for K is selected where both variance and bias are kept as low as possible[29, 33]. A good rule of thumb for selecting K is setting K to $\sqrt{n}$ where n is the number of data points in the training data[55].

The advantages of KNN are that it is simple and intuitive. It provides good results when enough training samples are provided and can be applied to any type of distribution. However it is hard to choose a correct value for K and with a large number of p KNN can become computationally very heavy and provide weak results, this is called the curse of dimensionality. Also KNN needs a very large number of training samples to provide accurate results[55].

7.2.3. Decision trees

Decision trees can be used for both classification and regression. The research is focussed on estimation thus regression trees are discussed, an example is shown in Figure 7.1. When using regression trees the predictor space, the space where the training set is mapped, is divided into non overlapping regions. To make a prediction on a test observation, the mean of the training observations of the region to which the test observation belongs is taken. Regression trees consist of a number of splitting rules which divide the current predictor space, which could be a subset of the entire predictor space, into 2 regions. This splitting is done based on a single variable. When the method is finished the predictor space is split in j regions. These regions are called the terminal nodes or leaves of the tree. Points along which regions are split are called internal nodes and internal nodes are connected to other internal nodes or leaves via branches. The first node of the tree is called the root. The described methodology is very likely to overfit the model. Therefore trees are pruned, leaves are removed to reduce the complexity of the model[29].

Trees are very easy to explain, easily interpreted and provide a nice graphical representation. Also trees are believed to better mimic human decision making than regression or classification approaches. Another advantage of decision trees is that trees are able to handle qualitative input. However as already mentioned they are very likely to overfit the model. Also trees are not as accurate as regression approaches and trees are not robust, which means that a small change in the training set can cause large changes in the final tree[29].

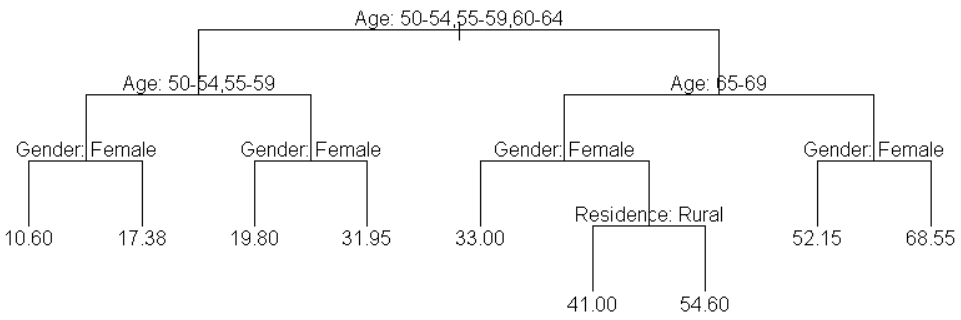


Figure 7.1: Example of a regression tree from[45]

7.2.4. Support Vector Machines

Support Vector Machines(SVMs) are one of the best ”out-of-the-box” approaches in machine learning, however they are based on binary classification which means that the output can only have two distinct values. SVMs make is possible to translate linear classification into non-linear classification by using kernels. The SVM tries to divide the training set into two sets of points. How these points are divided is somewhat unclear. The SVM works as a black box, some input is provided and the SVM gives an output. SVMs can be expanded to classification for $K > 2$ classes[29].

An extension of SVMs is Support Vector Regression(SVR)[29]. With SVR the input is mapped onto a p-dimensional space Φ. For the images of the points on the map Φ the dot products are computed. This corresponds to evaluating the kernel function of the SVM. The dot products are multiplied by a coefficient α_p and added up plus a constant term b. This produces the output of SVR[69].

Although SVM and SVR are one of the best ”out-of-the-box” approaches they are also hard to understand since they are based on a black box principle.

7.2.5. Neural Networks

Neural networks(NNs) make predictions based on neurons that are connected by weighted links. An example of a NN is shown in Figure 7.2. This is a multilayer perceptron. The performance of these NNs is remarkably high. It is advised to use normalized input on the interval $[-1, 1]$ with NNs. Neurons are the basic units of a NN. Based on the sum of weighted input the response of a neuron is calculated using a transfer function[33]:

$$f(\Sigma) = \frac{1}{1 + e^{-\Sigma}} \tag{7.2}$$

Neurons are arranged in an output layers and in the hidden layers. A NN can have one or more hidden layers, although in most cases the amount of hidden layers will not exceed three. Neurons do not communicate on the same layers but are fully connected to the adjacent layer, meaning one neuron is connected to every neuron in the adjacent layer. Every link has an associated weight. Data is forward propagated through a NN. The weight are set by back propagating the error through the NN and setting the weights accordingly[33].

Although Neural Networks can have a very high performance, they are computationally expensive and have a high risk of overfitting the model. Also configuring the network is very difficult and may take a lot of time[33].

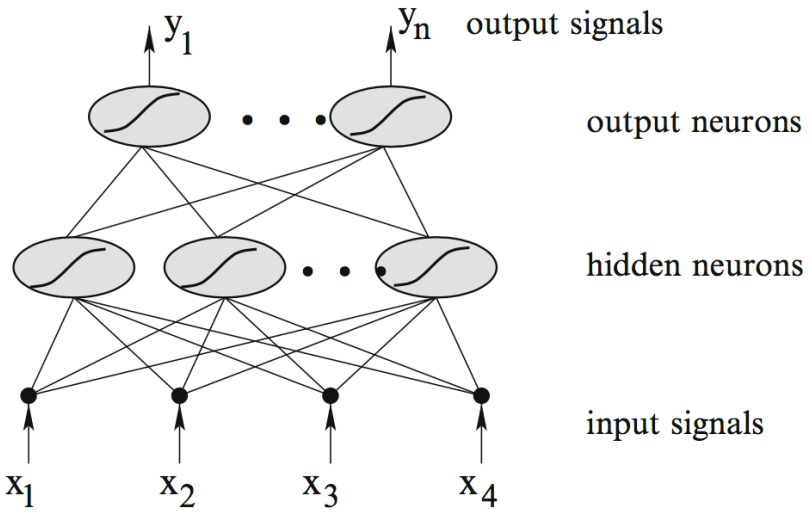


Figure 7.2: Example of a Neural Network[33]

7.3. Method selection

To select the prediction method some requirements are in line with the research. As discussed in the first section multiple prediction models are created, one for each route. Therefore it is important that the selected method is easy to use, easy to configure and computationally cheap. Furthermore because a proof of concept is provided, a method that is easy to understand is needed. Another constraint on the selection of the prediction method is time. Since limited time is available, not a lot of time is put into configuring a network. Therefore the requirement to have a prediction method that is easy to configure is very important. The discussed methods are teseted on these requirement in Table 7.1.

Table 7.1: Fit to requirements of prediction methods

Prediction method	Easy to use	Easy to configure	Computationally cheap	Easy to understand
Multiple linear regression	X	X	X	X
KNN regression	X	X	X[1]	X
Regression trees	X	X	X	X
SVR	X	X		
NN	X			

Because a proof of concept is provided, the best performance available is not needed and therefore this is not a requirement. Due to time limitations is is needed to quickly compute results that are easy to understand and therefore Support Vector Regression or Neural Networks are not used. Also since Regression trees have a big risk of overfitting and are very dependent on the input, these are deemd not suited for the research. A choice between linear regression and KNN regression is needed. Since a linear relationship may not be present in the data, KNN regression is chosen for prediction, if a linear relationship is present K can be set to such a value that it mimics a linear relationship.

Although KNN does not have the best performance available, this method is chosen because it is able to generate results very quick if the amount of predictors is low and in the research time is limited. In Section 5.7 7 predictors are identified so the amount of predictors is low and therefore using KNN will be quicker then using NNs that take a lot of time to configure per model. And since multiple models are constructed, using NNs will lead to using a lot of time to configure every model. With KNN only the amount of neighbors needs to be decided upon, so results are generated quickly. Setting K to the best possible variable is not easy, but can be quickly researched upon by testing the performance of the model for different values of K.

Another possible problem with using KNN is the low amount of training data available in the research. KNN requires a lot of training data to produce accurate predictions. However this also holds to a lesser degree for every other prediction method in general, so the results of the prediction will always be negatively influenced by the small amount of training data. And as already mentioned a proof of concept is provided which means that the use of the framework is demonstrated and the framework is not assessed on its results.

[1] While using a small number of predictors

7.4. ETA prediction

K-Nearest Neighbor(KNN) regression is selected for predictions so the data should be prered, the model trained and its performance briefly evaluated. Because of the low quality of the data not a lot of recurring routes are left in the dataset. Therefore framework is demonstrated with the help of the 6 most visited routes in the dataset.

1. HAMBURG -> ROTTERDAM
2. DUBLIN -> ROTTERDAM
3. MOERDIJK -> ROTTERDAM
4. BREMERHAVEN -> HELSINGBORG -> GOTEBORG -> ROTTERDAM
5. FELIXSTOWE -> HAMBURG -> ROTTERDAM
6. DUBLIN -> SOUTHAMPTON -> ROTTERDAM

7.4.1. Preparation

To use KNN regression a training and a test set are needed. So the data needs to be split. However because predictions are based on routes a training set is needed for each route. To select the observations that visit the six most visited routes the data frame containing the route of each vessel per voyage and the route database is used. If the route of an observation is in the top six most visited routes the imo and voyage id of this observation is added to a new dataset, ”useful routes”.

Cross validation

To estimate the performance of prediction algorithms, validation is used. Two types of validation exist: simple validation and cross-validation. To do any type of validation, the data is split in a training sample, used for training the model, and a test set used for validating the performance. With simple validation, also known as hold-out validation, a single split is made into a training set that consist 70% of the data and a test set that consists 30% of the data. With cross-validation the data is split one or multiple times[3, 67]. One of the main reasons to use cross-validation is a small amount of data that is available to partition, when a small amount is available significant modelling or testing capabilities are lost, in these cases cross-validation is a powerful technique to use[67]. Other strong characteristics of cross-validation are that it can be aplied to all prediction methods, avoids overfitting and reduces variability by averaging the errors over the rounds[3, 67]. Since cross-validation is beter suited to evaluate the performance than simple validation, the performance in this research is assessed by using cross-validation. Some classical cross-validation methods are discussed below.

Leave-one-out Two type of splitting schemes exist for cross-validation: exhaustive data splitting, that considers all training set of size n_t, and partial data splitting. Leave-one-out is an exhaustive data splitting scheme. The created training sets are of the size $n_t = n - 1$, where for every round a single data point is successively left out from the training set and used for validation[3]. In this research that translates into leaving out one voyage per route.

Leave-p-out Leave-p-out is similar to Leave-one-out but $n_t = n - p$, where $p > 1$ otherwise it would be Leave-one-out. Every possible subset of p datapoints, p voyages per route in this research, are succesively left out and used for validation[3].

V-fold cross-validation V-fold cross-validation is a scheme of partial data splitting. Before training the data set is split in V subsets of data of the size $\frac{n}{V}$. V rounds are performed with a training set of size $V - 1$ and each subsample is the test set exactly once[3]. In this research simply splitting the data set could cause succesive observations of a single voyage to be seperated into different subsets, so subsets of voyages per route should be created.
A small amount of data per route is available and using Leave-one-out makes it possible to use the biggest possible training sets while maintaining the benefits of cross-validation. Using bigger training sets provides better predictions, so Leave one-out cross-validation is used in the research. Other methods would lead to smaller training sets and are therefore not used.

Data splitting

Since performance is assessed by using Leave-one-out cross-validation, the ith voyage per route from the usefull route dataframe is selected as the test route, the remaining routes are selected as training routes. A semi-join is made of the AIS database and training routes. This results in a general training set. To this set the routes for every observation are added by making a left join of this training set and the training routes. This data is split into different training set per route. To mimic real-time AIS messages the test sets are not seperated, so a semi join of the AIS database and test routes is made and arranged by time. These training sets and test set are used for predictions.

7.4.2. Training the model

In the general case a model needs to be trained before it can be used for prediction. However with KNN regression it depends on the package that is used in R. Because the FNN package is used, the model does not need to be trained before observations are passed from the test set to the model. A value for K needs to be set. The rule of thumb as introduced by Osadchy is used to set the initial value of K, where K needs to be set to the square root of the number of observations in the training set[55]. Since the framework uses multiple training sets the following formula for K is defined.

$$K_{route} = \sqrt{n_{route}} \tag{7.3}$$

Where K_{route} is rounded and n_{route} stand for the number of voyages in the training set of the relevant route. Furthermore a set of predictors is identified. As already discussed in Section 5.7 the set of variables presented in Table 7.2 is the set of the predictors:

Table 7.2: Predictors

Predictor	Type	Possible values
Latitude	Numeric	$-90°$ to $90°$
Longitude	Numeric	$-180°$ to $180°$
Draught	Numeric	0 to 255
Lenght of the vessel	Numeric	0 to ∞
Widht of the vessel	Numeric	0 to ∞
Estimated travel time	Numeric	$-\infty$ to ∞
Navigational status	Nominal	0 = under way using engine, 1 = at anchor, 2 = not under command, 3 = restricted maneuverability, 4 = constrained by her draught, 5 = moored, 6 = aground, 7 = engaged in fishing, 8 = under way sailing, 9 = reserved for future amendment of navigational status for vessels carrying DG, HS, or MP, or IMO hazard or pollutant category C, high speed craft (HSC), 10 = reserved for future amendment of navigational status for vessels carrying dangerous goods (DG), harmful substances (HS) or marine pollutants (MP), or IMO hazard or pollutant category A, wing in ground (WIG); 11 = power-driven vessel towing astern (regional use); 12 = power-driven vessel pushing ahead or towing alongside (regional use); 13 = reserved for future use, 14 = AIS-SART (active), MOB-AIS, EPIRB-AIS 15 = undefined = default (also used by AIS-SART, MOB- AIS and EPIRB-AIS under test)
Shipping line	Character	AL, APL, ASTRO, ATLANTIC, BBC, BF, BG, BOMAR, BOX, CAP SAN, CMA CGM, CONMAR, CONTI, COSCO, CSCL, EXPRESS, DS, E.R., ECL, EM, EVERGREEN, FRISIA, BRIDGE, HANJIN, HANSA, HS, HYUNDAI, ICE, JORK, JPO, SCHEPERS, MAERSK, MAX, MOL, MSC, MV, NORDIC, NYK, OOCL, OPDR, PHOENIX, THALASSA, UNKWON, WES, WILSON, XIN, YM and ZIM

7.4.3. Running example

In this section a simplified running example of the ETA prediction algorithm is shown. The Table 7.3 shows a simplified route database containing only two routes, as could be created by the first three steps of the framework as presented in Figure 6.1. The following table is the vessel database where the vessels under observation are stored. This is followed by a table containing a simplified AIS message. Following this message are tables containing the updated vessel database and tables containing the results of the identifications, predictions and the next AIS message. This running example is an expansion of the running example showed in Subsection 6.3.3. In this example the predicted times to arrival are added. The number that are shown lie in the possible range but are not the result of real predictions.

Table 7.3: Route Database

Route	IMO Numbers
Dublin → Rotterdam	1
Dublin → Southampton → Rotterdam	1, 2

Table 7.4: Vessel database

Imo number	Voyage

Table 7.5: first AIS message

Imo number	Destination	Based on route	Predicted time to arrival
1	Dublin		

Table 7.6: Vessel database

Imo number	Voyage
1	Dublin

Table 7.7: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
1	Dublin	Dublin → Rotterdam	100 hours
		Dublin → Southampton → Rotterdam	150 hours
2	Dublin		

Table 7.8: Vessel database

Imo number	Voyage
1	Dublin
2	Dublin

Table 7.9: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
2	Dublin	Dublin → Southampton → Rotterdam	140 hours
1	Southampton		

Table 7.10: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin

Table 7.11: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
1	Southampton	Dublin → Southampton → Rotterdam	90 hours
3	Dublin		

Table 7.12: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin
3	Dublin

Table 7.13: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
3	Dublin	Dublin → Rotterdam	80 hours
		Dublin → Southampton → Rotterdam	130 hours
2	Rotterdam		

Table 7.14: Vessel database

Imo number	Voyage
1	Dublin → Southampton
2	Dublin → Rotterdam
3	Dublin

Table 7.15: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
2	Rotterdam	Dublin → Rotterdam	48 hours
1	Rotterdam		

Table 7.16: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin

Table 7.17: Result & next AIS message

Imo number	Destination	Based on route	Predicted time to arrival
1	Rotterdam	Dublin → Southampton → Rotterdam	36 hours
3	Dublin		

Table 7.18: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin

Table 7.19: Result & Last AIS message

Imo number	Destination	Based on route	Predicted time to arrival
3	Dublin	Dublin → Rotterdam	70 hours
		Dublin → Southampton → Rotterdam	120 hours
3	Rotterdam		

Table 7.20: Vessel database

Imo number	Voyage
1	Dublin → Southampton → Rotterdam
2	Dublin → Rotterdam
3	Dublin → Rotterdam

Table 7.21: Result

Imo number	Destination	Based on route	Predicted time to arrival
3	Rotterdam	Dublin → Rotterdam	50 hours

7.4.4. Model evaluation

In this section the performance of the model is evaluated. The influence of using routes, different amounts of neighbours and different predictors are evaluted. Also outliers are identified in the next section and removed from the data.

K

In this section the performance of the algorithm is tested with value $K_{route} = \sqrt{n_{route}}$. Where K_{route} is rounded and n_{route} stands for the number of voyages in the relevant training set. The results with outliers are shown in Figure 7.3. The blue lines show the estimation errors of the vessel and green lines show the prediction errors. Light lines show the average error and dark lines the median error value. All actual hours to arrival are rounded for the readability of the graphs. So the error at 1 hour to arrival actually covers the average errors and median value for 0,5 to 1,49 hour to arrival.

Figure 7.3 shows that the performance for the two routes starting in Dublin is almost equal to the best guess of the captain. The errors and median values are more or less equal. But the Dublin -> Rotterdam route has a small spike in the median prediction error between hours 100 and 150. This may be smoothed out by using different values for the amount of neighbours. But first the other routes are discussed since these show that the performance is in some cases much better and in other cases much worse. The predictions of these routes are more closely assessed.

Bremerhaven From 175 hours to arrival the prediction error suddenly skyrockets. A closer look at the predictions shows that a single voyage has a lot of time at anchor due to christmas and New Years Eve. Therefore this voyage is the only route that took longer than 175 hours and the predictions are based on voyags with shorter travel times. Therefore is such a huge error present in the predictions. Since this voyage is unique and no similar patterns have been found this voyage is deleted from the dataset.

Felixstowe The route starting in Felixstowe shows somewhat similar results as the Bremerhaven route. Suddenly a big prediction error from a specific amount of hours to arrival. This error is caused since only a single voyage has a travel time that is over 200 hours. So predictions for this voyage are based on shorter voyages and thus predictions with huge errors are created.

Hamburg At the Hamburg route huge estimation errors are present. Due to the data quality of the ETA attribute in AIS messages, some voyages with estimated travel times of over 500 hours are used and thus bias the estimated travel times. Also the median prediction error and the average prediction error have a big difference above 200 hours. A vessel was found that actually travelled the Felixstowe route, but due to a very early change in destination this was not handled by the pre-processing framework. So this voyage is manually manipulated. Also around 100 hours an increase in the errors is observed, while assessing the predictions it is noticed that most Hamburg to Rotterdam voyages take roughly 80 hours. When the voyage took longer the vessel actually started in Rotterdam, sailed to Hamburg and back. Since the positional data of a vessel entering and leaving the Port of Rotterdam is similar, this hampers the performance. A possible solution would be to incorporate the course in the set of predictors. Also a voyage was observed that was laying at anchor at sea for a very long time before entering the Port of Hamburg. This ship probably was empty and not in use, so this voyage is removed from the data set.

Moerdijk The voyage from Moerdijk to Rotterdam takes roughly 5 hours as is observed in the predictions. However voyages of over 150 hours are observed. Four voyages have been identified where vessels were moored in Moerdijk for multiple days, so the vessel was probably not in use but kept sending AIS messages. These voyages are removed from the dataset. As is the case with the Hamburg route, also a lot of vessel start in Rotterdam with their voyage. So again incorporating the course may improve the predictions.

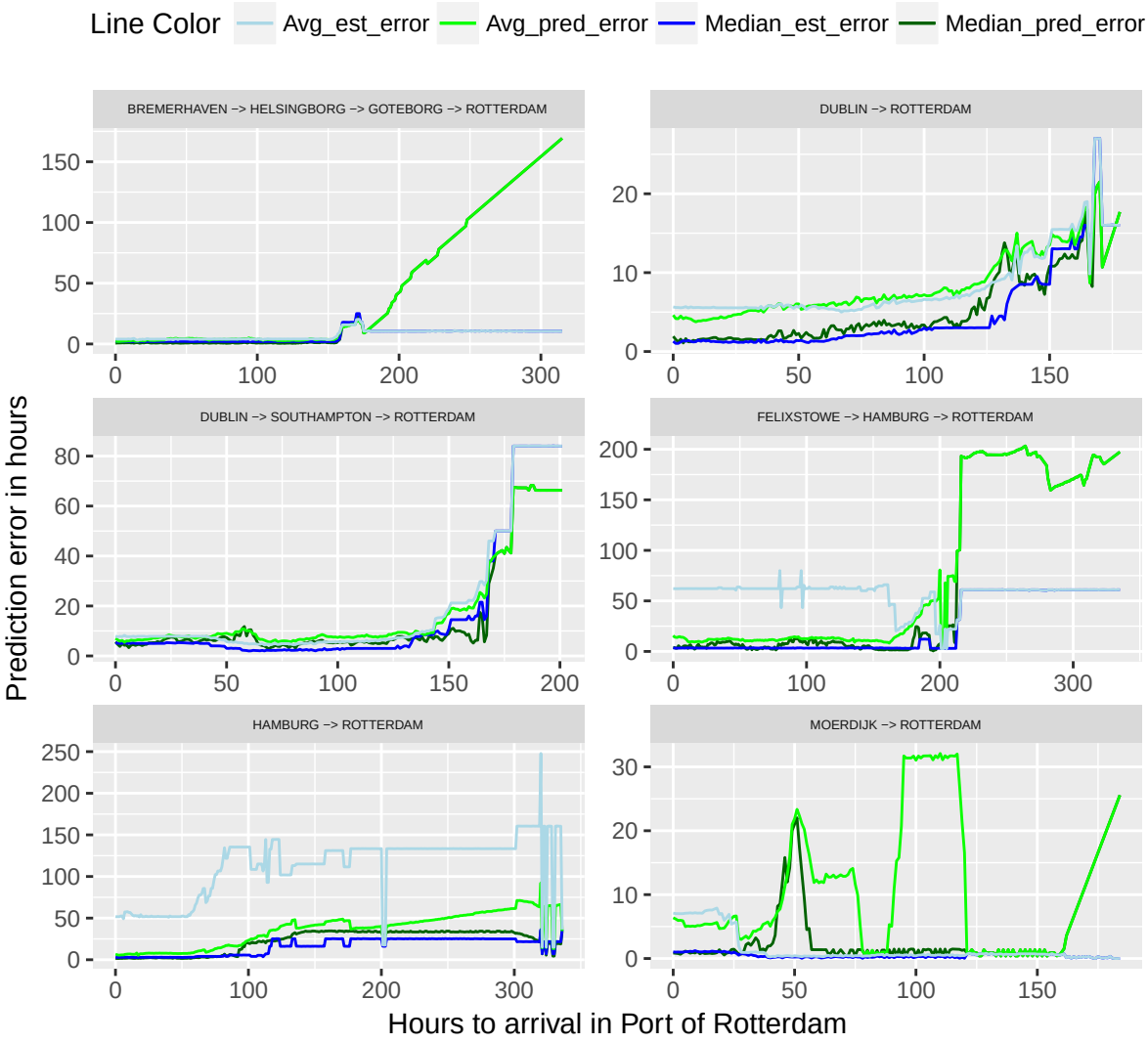


Figure 7.3: Original data set, cross validated with K

Figure 7.4 shows the results of the predictions with the modified dataset. The Bremerhaven route now shows better performance that is also equal to the best guesses of the vessels. The Felixstowe route shows larger estimation errors, this is due to the fact that the vessel, that was added from the Hamburg route, has large estimation errors that are even over 900 hours. For the Hamburg route the prediction errors are roughly half a small above 100 hours to arrival or even smaller. For the Moerdijk route the performance is actually worse because still a few voyages remain that have longer travel times than most voyages. So the predictions are based on shorter routes but from 20 hours (the time most voyages take) and less the performance is equal to the best guess of the vessels crew. In general the predictions have improved so this data set is used to compare different configurations. First the influence of the amount of neighbours is assed.

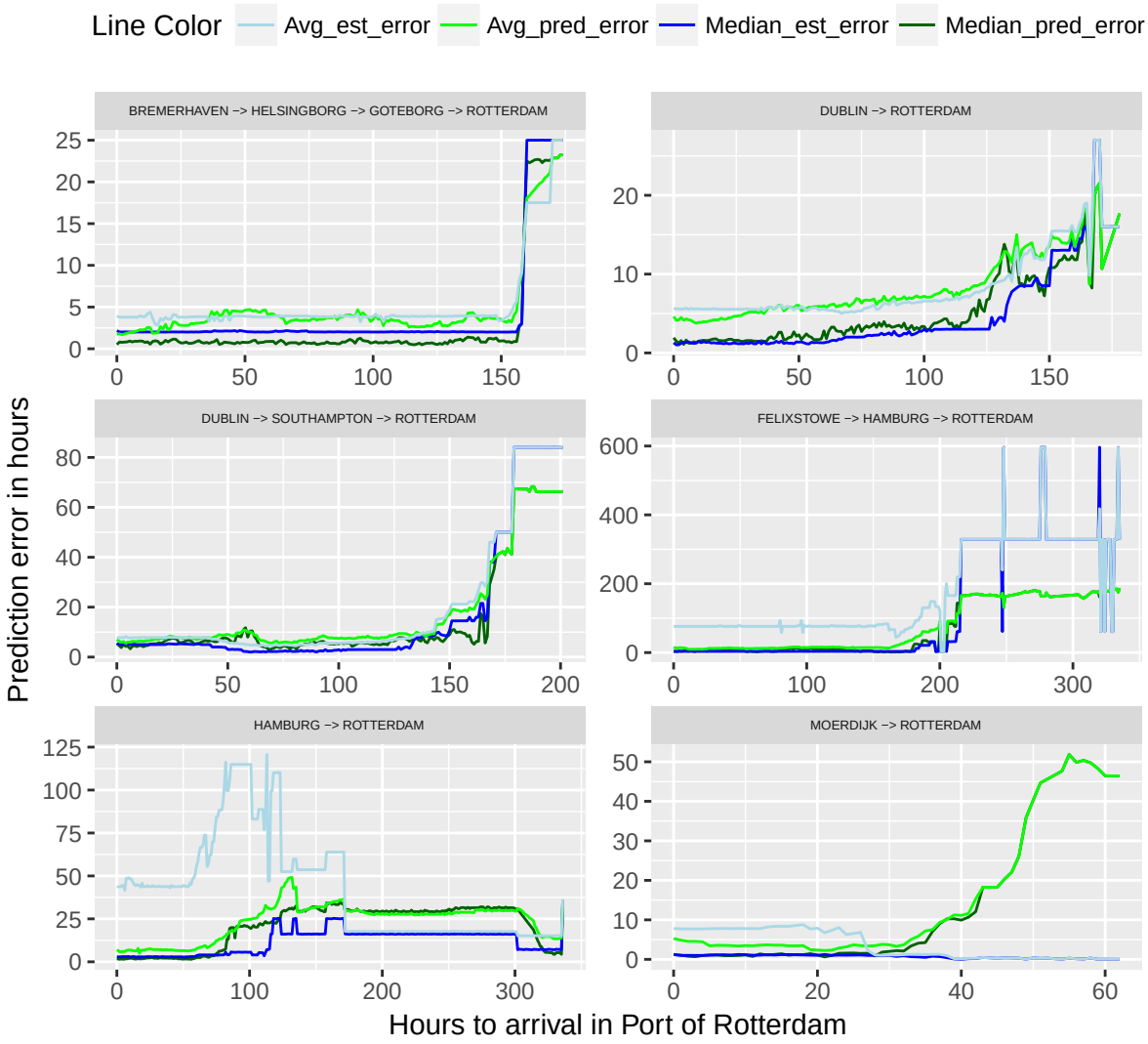


Figure 7.4: Manipulated data set, cross validated with K

Amount of neighbours

In this section the influence of the amount of neighbours is assessed. Figure 7.5 shows the results for $K+2$ so $K_{route} = \sqrt{n_{route}} + 2$ and Figure 7.6 shows the results for $K-2$ so $K_{route} = \sqrt{n_{route}} + 2$. Using different values for the amount of neighbours impacts the volatility of the predictions. Using more neighbours will create less volatile predictions, so for further assesments the amount of neighbours used is $K+2$

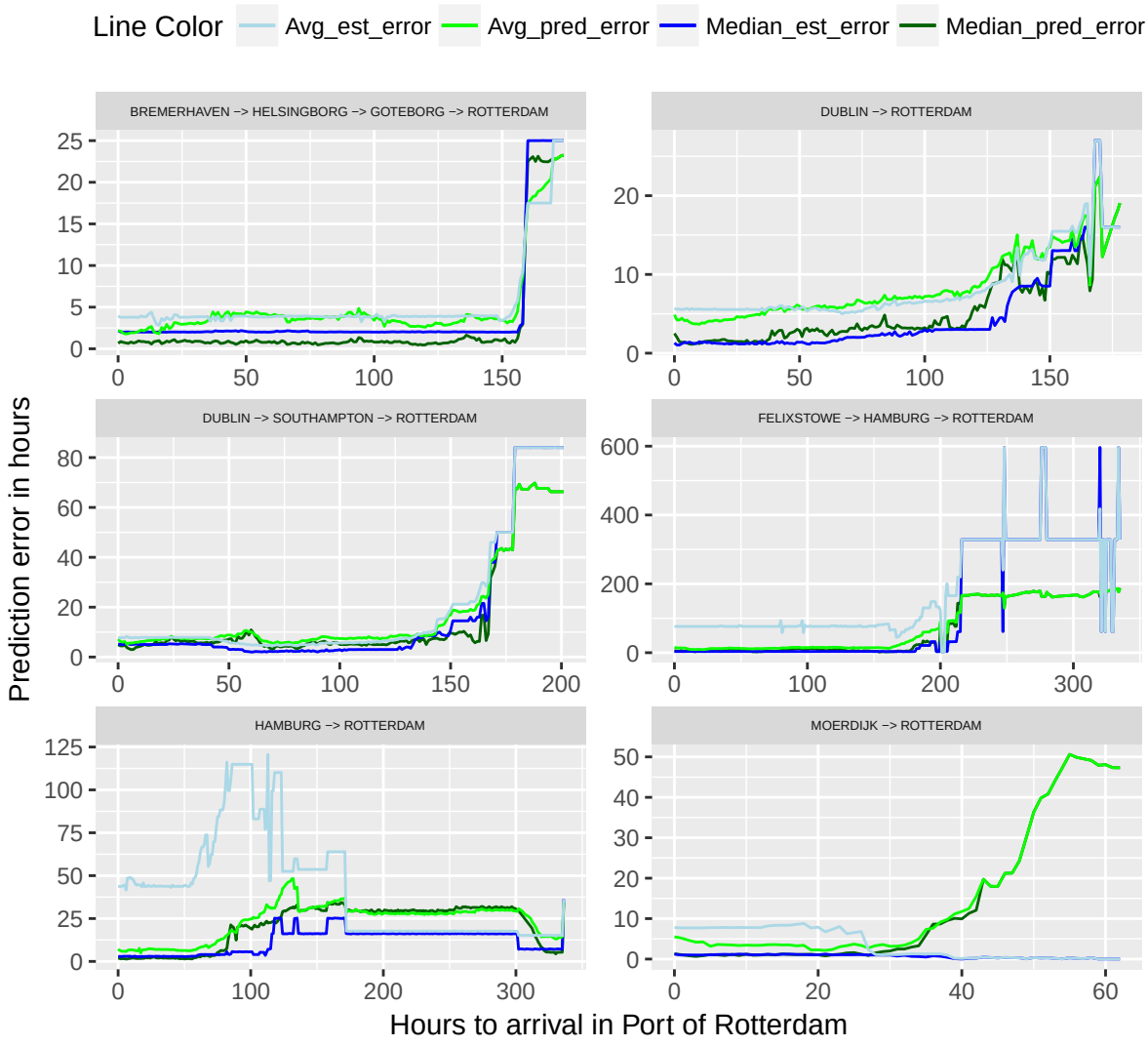


Figure 7.5: Cross validated with K+2

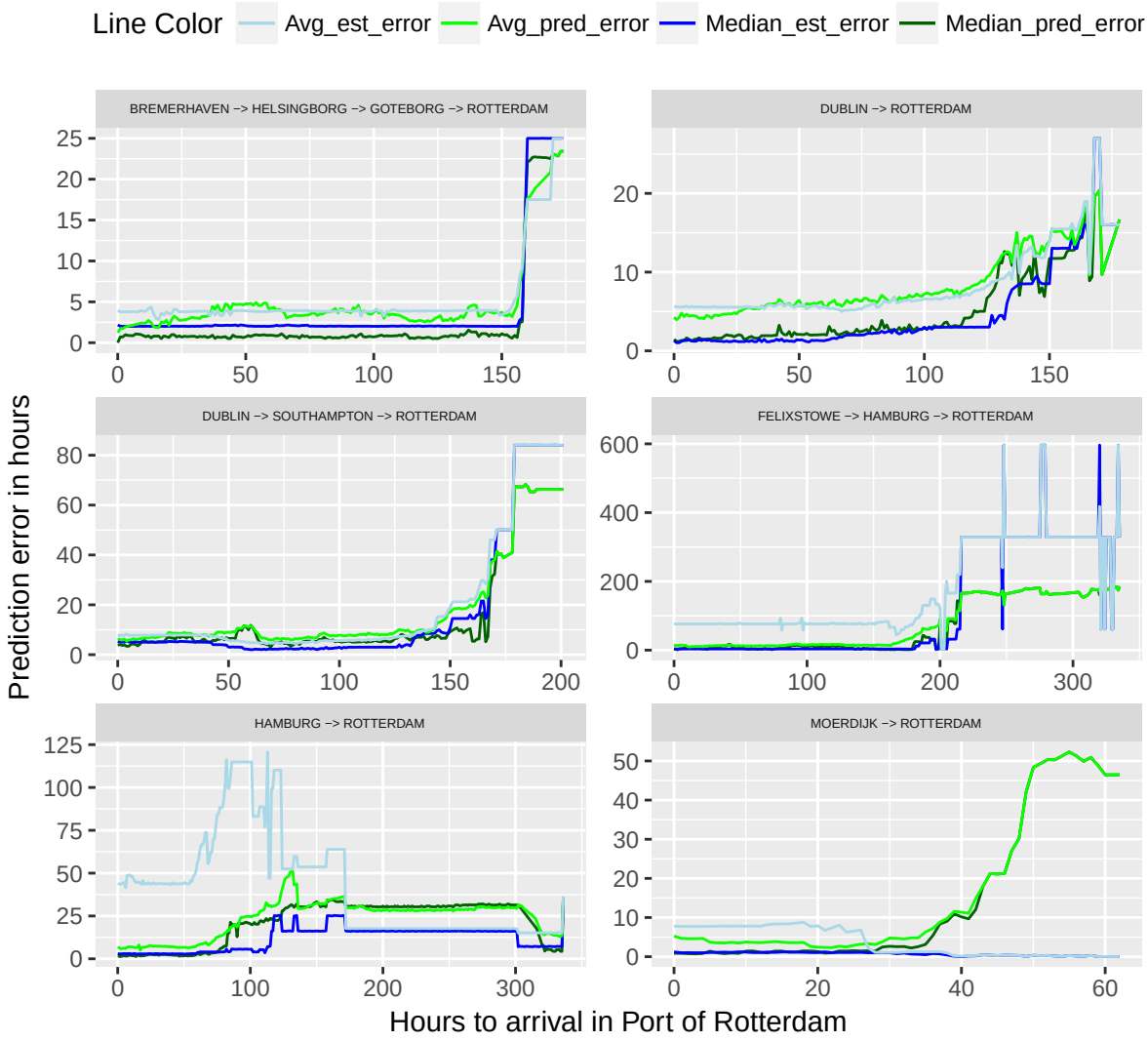


Figure 7.6: Cross validated with K-2

No estimated time to arrival

Using the estimated time to arrival is identified as one of the most important predictors by Parolas[58]. To check this claim predictions have been produced without using the estimated time to arrival. The results are shown in Figure 7.7. The graphs clearly shows that both average as median prediction errors are much larger is most cases. However when not a lot of vessels with the same estimated time to arrival are available in the dataset, the predictions actually improve. For example, in the Felixstowe route the prediction error above 200 hours to arrival is roughly 175 hours when the estimated time to arrival is incorporated and the prediction error is under 100 hours when it is not incorporated. Similar conclusions can be drawn for the Moerdijk route. So using the different sets of predictors for every route may improve the performance of the algorithm.

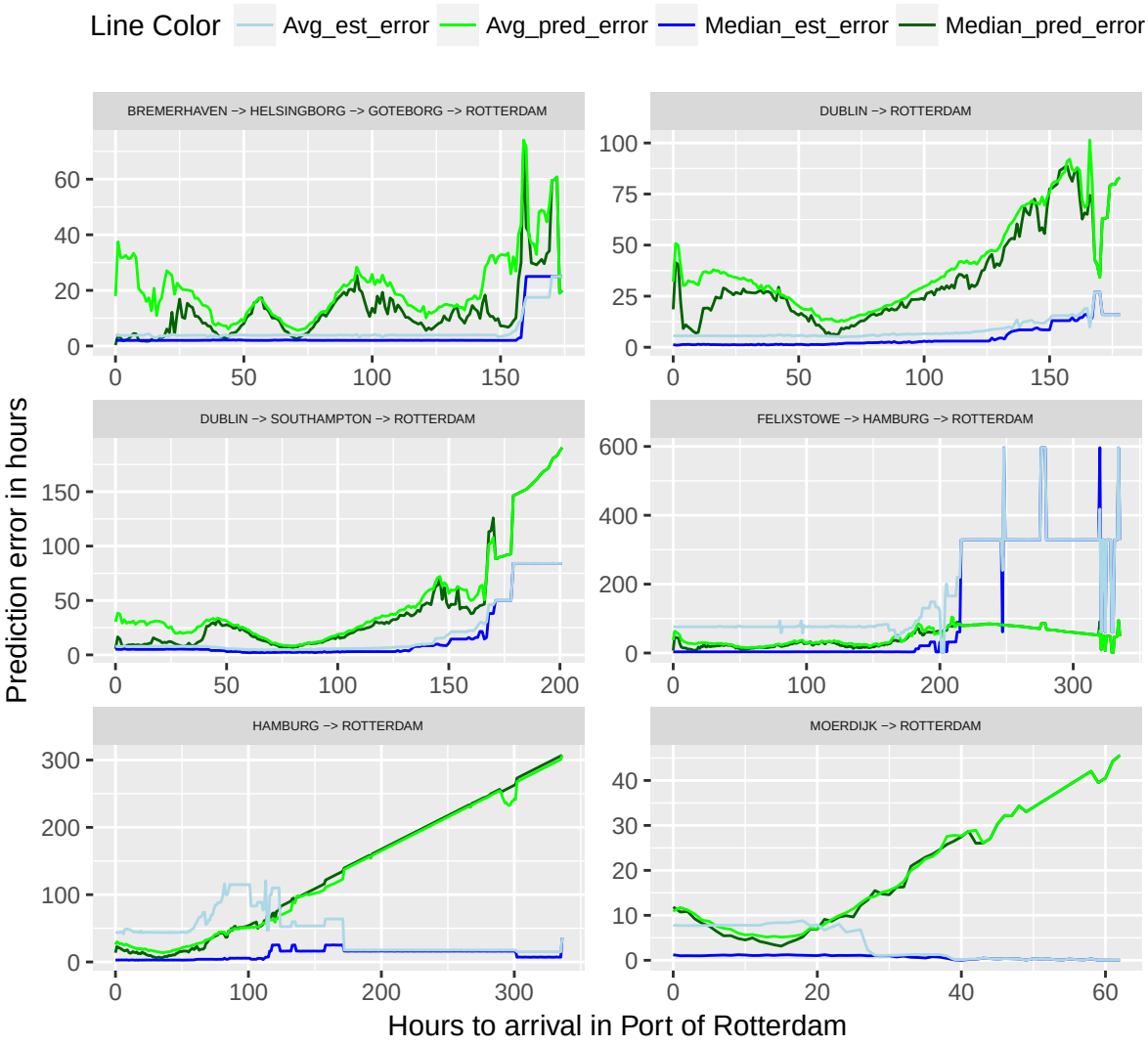


Figure 7.7: Cross validated with K+2 and no ETA as predictor

Using course

As identified during the thorough assesment of the predictions, some vessels start their voyage in Rotterdam so their positional data is similar to the positional data of vessels that are at the end of their voyage. Therefore incorporating the course may improve the predictions, especially for the Hamburg and Moerdijk route. The results of the predictions with the course as a predictor are shown in Figure 7.8. When comparing these results to Figure 7.5 no significant differences are observed. So using course makes no differences. This is probably the result of using the estimated time to arrival as a predictor. The estimated time to arrival is much bigger when a vessel start in Rotterdan in comparison to arriving in Rotterdam. So vessel that leave and arrive in Rotterdam are not considered to be close neighbours. However when the amount of neighbours is further increased, the incorporation of course may have benefits.

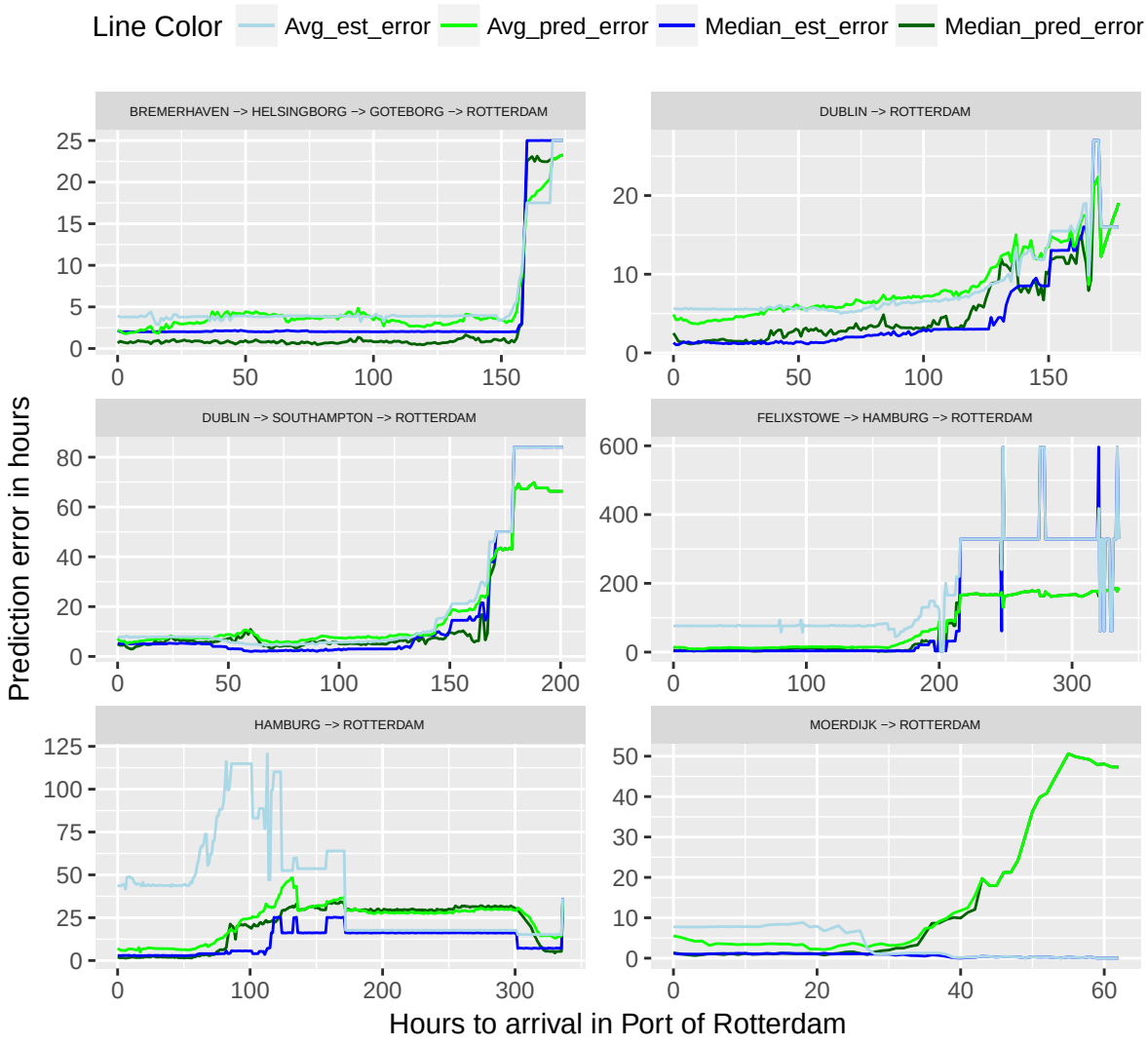


Figure 7.8: Cross validated with K+2 and Course as predictor

Not using routes

One of the key assumptions that is made in this research is that using routes increases the accuracy of the predictions. Figure 7.9 shows the results of the algorithm when a general model is used. In this case the average prediction error is smaller in general in comparison to the average estimation error. But the estimation error is biased due to the probably faulty very large estimated travel times. When the estimated travel times are not used, as shown in Figure 7.10, the performance is much worse. This again supports the claim by Parolas that estimated travel times are the most important predictor. The predictions above 200 hours to arrival are highly volatile but this is because not a lot of long vessels are present in the data set. So a small gap in one of those voyages has a large influence on the predictions.

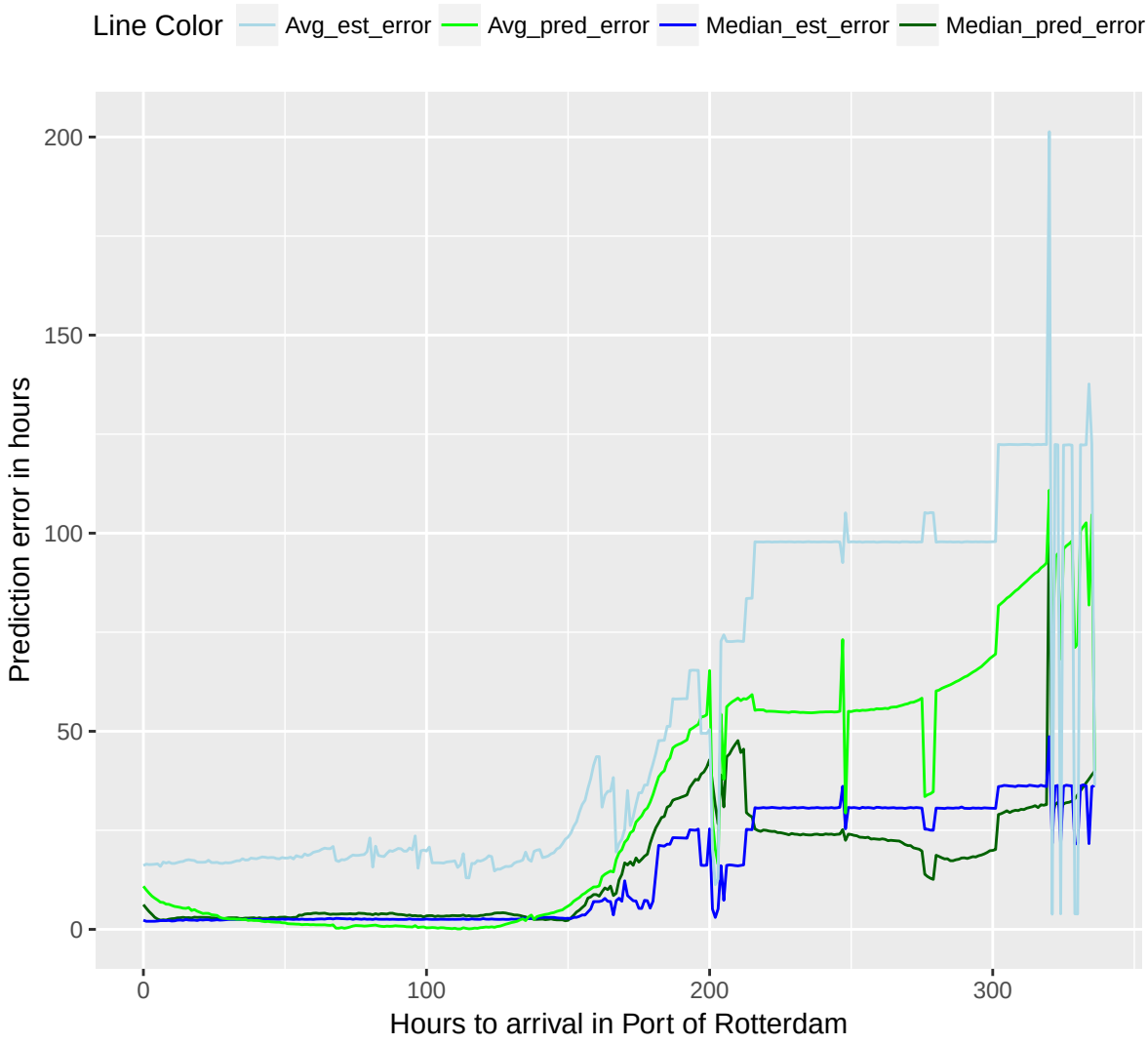


Figure 7.9: Not using routes, cross validated with K+2

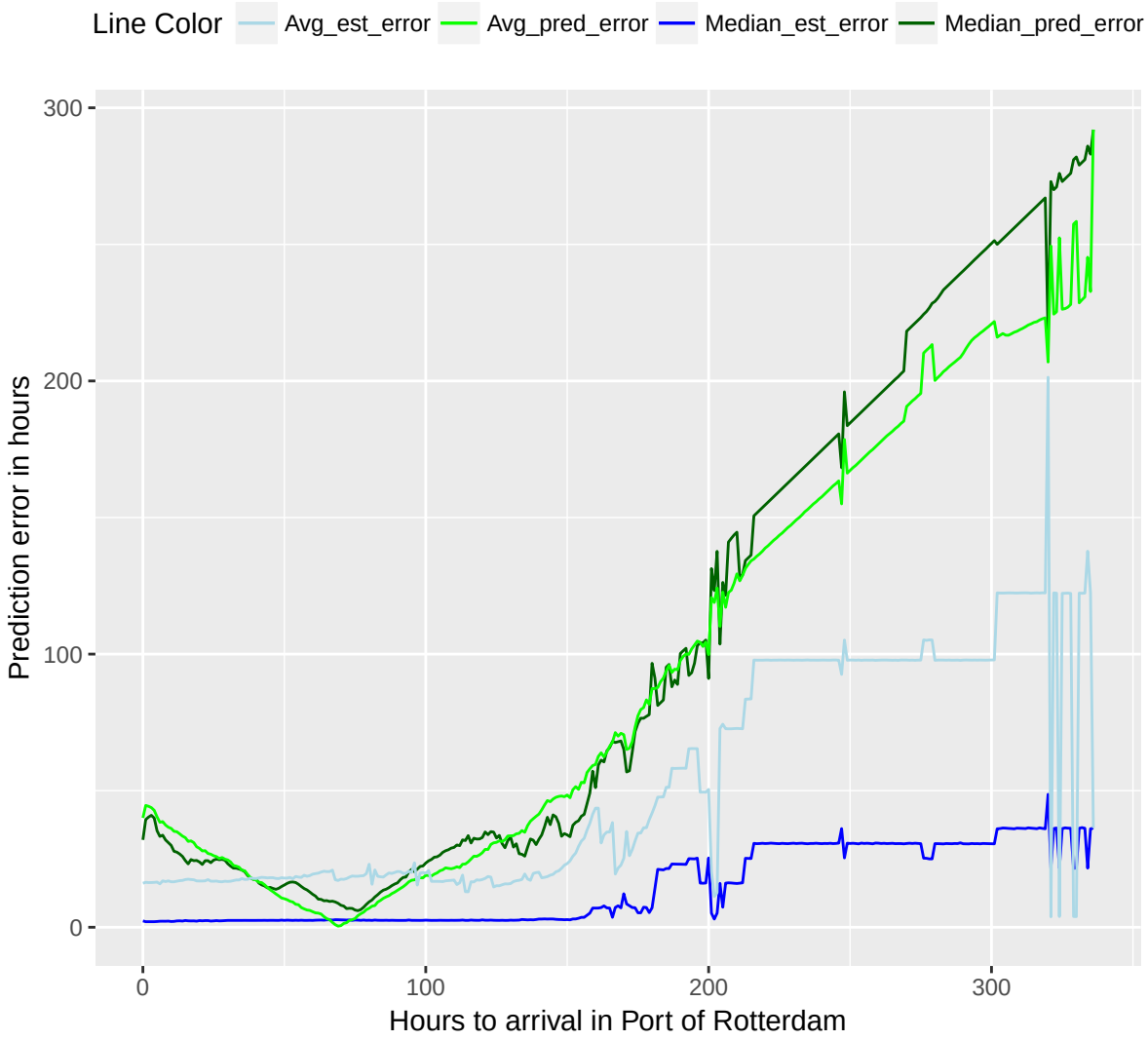


Figure 7.10: Not using routes and ETA, cross validated with K+2

The maximum average error is roughly 100 hours, but since the errors are averaged over more voyages this average may be lower then with the use of routes, while the error per route does not reduce. So the minimal and maximal error are also shown in Figure 7.11. This shows that the maximal errors above 200 hours are similar to the average errors in the Felixstowe route. So although the performance seems better, this may not be the case. The big errors due to a lack of data of specific routes at specific points are just averaged out. Also the predictions are based on more data and therefore may be better. So to provide a definitive answer if routes have a positive impact on the performance, the research needs to be repeated with a bigger data set of a better data quality. Therefore the hypothesis is posed that using a bigger data set of a better data quality has a positive impact on using routes in ETA predictions. For this hypothesis further research is needed.

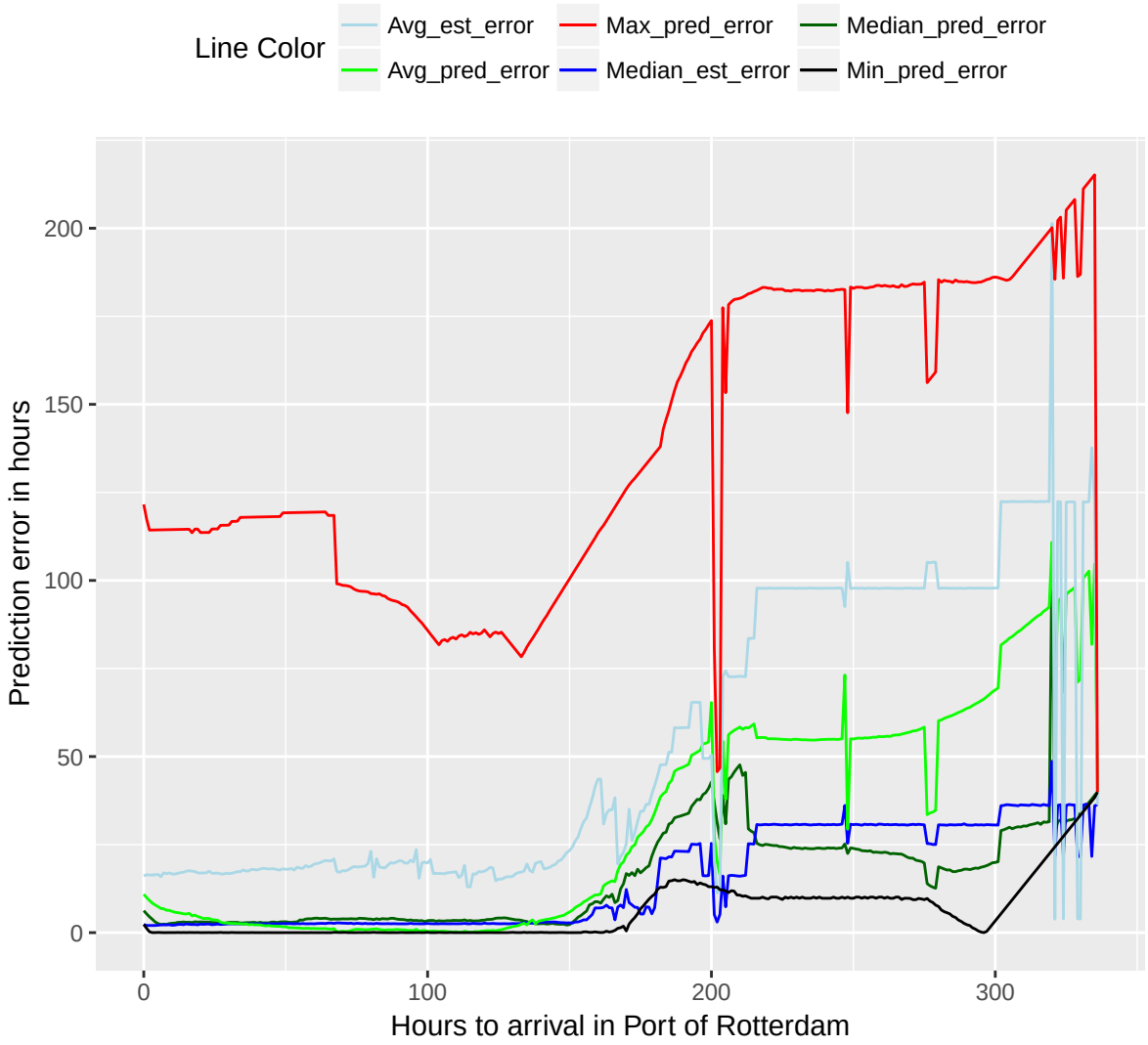


Figure 7.11: Not using routes, cross validated with K+2

7.5. Information representation

Since the framework does not select a single route to do predictions but used the set of possible routes. All possible predictions are presented to the port with additional information, the vessel and route used for specific prediction. It is then up to the port to decide which prediction to use in the planning. Extra information can be gathered by the port or past experiences can be used for selecting a prediction. Selecting a single route and providing a single predictions is proposed for further research.

7.6. Conclusion

In this chapter the research question *How can the ETA of a vessel be predicted with the use of pre-processed AIS data and route identification?* is answered. Routes can be used to select prediction models that are based on specific routes, however it is unclear if this improves the performance of the predictions and further research is needed. But in general using machine learning makes it possible to make prediction that are of equal quality in comparison to the best guess of the vessel's crew. Algorithm 7.2 shows a framework to do model selection based on routes and predict the ETA of a vessel based on a pre-processed AIS message. The ETA as communicated via the AIS message is identified as an important predictor in most cases. However in some cases this hampered the predictions. So further research is warranted that assesses which predictors to use for which routes.

Algorithm 7.2 Framework for ETA prediction

Input: AIS database with standardized destinations, database with possible routes and vessels that travelled those routes

1: Create training set for each route and test set from AIS database
2: Create prediction model for each route
3: Read AIS message from test set
4: Identify possible set of routes for vessel *# Algorithm 6.3*
5: Select prediction models based on possible set of routes
6: Do predictions

Output: Prediction for each possible route

8

Stakeholder analysis

Now a methodology to predict the ETA of a container vessel is constructed, the relevant stakeholders are discussed. The research question "What is the added value of the proposed algorithms and framework for stakeholders in the supply chain and how can stakeholders influence the data quality of AIS messages?" is answered in this chapter. First the relevant stakeholders are briefly introduced, this is followed by showing the relevant value for these stakeholders. In the next section of the chapter the power of a stakeholder to improve the quality of AIS data is discussed.

8.1. Stakeholders in the supply chain

Several stakeholders are involved in a containerized supply chain, Figure 8.1 provides an overview of these stakeholders. The stakeholders can be divided in two subgroups: Governance stakeholders and operational stakeholders. Every problem, that is a consequence of the uncertainty regarding the arrival time of a container vessel, is on the operational level. Therefore only operational stakeholders are discussed in this section. Furthermore ETA prediction is for incoming containers only stakeholders at import side of the supply chain are relevant. At the import side Parolas[58] has identified the Shipping line, Terminal operators, Forwarders and hinterland carriers, the port and the consignee as relevant stakeholders for an ETA prediction tool. Dobrkovic et al[12] identified the shipping line, terminal operator and the forwarders and hinterland carriers as relevant stakeholders. The shipping line, terminal operators, forwarders and hinterland carriers and the port are discussed.The consignee is neglected bcause the consignee receives the goods and the only added value for them is more information regarding the arrival of their package. It does not add value or changes operations in the supply chain. The relevent stakeholders are briefly discussed in this section.

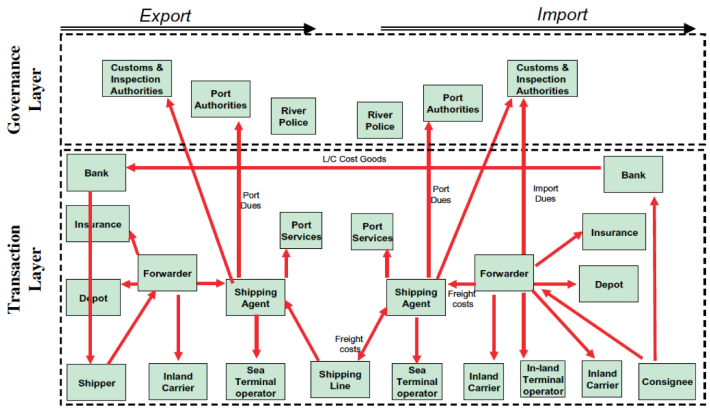


Figure 8.1: Overview of the stakeholders in a containerized supply chain. Image from van Oosterhout, Zielinksi & Tan[80]

8.1.1. Shipping line

Shipping lines are used to transport the containers from an origin sea port to a destination sea port. The market for shipping lines is very competitive, a lot of competing companies are in the market without a significant market share. Therefore the market can be characterized as a perfect competition where prices depend on supply and demand[71]. Because of an abundance of supply and a lack of demand the market is in a downturn[76]. As a consequence shipping lines try to minimize their cost. This can be done for example by operating the vessel at a speed as low as possible to reduce fuel consumption, since fuel accounts for roughly 60% of the vessel's expenses[71]. Operating the vessels at these varying speeds is called slow steaming[58]. The main interest of the shipping line is providing a reliable connection to the destination with on time deliveries at minimal operating cost[58].

8.1.2. Terminal operators

As already mentioned earlier in Chapter 2 a container terminal consists of three parts. The vessel operation area were containers are loaded onto or unloaded from vessels, the yard where containers are stored and the hinterland operations area where containers are loaded onto or unloaded from either barge, train or truck[18, 46, 58]. The yard facilitates the decoupling between the shipping operations and the hinterland operations[86]. The terminal operator is responsible for the planning off all the activities in the container terminal. So they perform the following planning operations[58]:

1. Berth planning, planning the berths and time slots for the container vessels.
2. Yard planning, decide which container is stored where in the yard.
3. Vessel planning, planning the order of loading and unloading the vessel while maintaining stability and safety of the vessel.
4. Resource allocation, allocate the required manpower and resource to carry out the operations.

Their main goal is to come up with a planning that requires as little changes as possible, since these changes will lead to an increase in costs.

8.1.3. Forwarders and hinterland carriers

Forwarders connect the deep sea terminals to inland terminals. Containers are received by the forwarder at the deep sea terminal and are assigned to a truck, barge or train of a hinterland carrier to be transported to the inland terminal. Because competition increased between ports, it is of vital important to provide cost-efficient and reliable networks that can be used to transport the container to the hinterland. So the interest of the forwarders and carriers are providing cost-efficient, safe and reliable services of the hinterland network[58].

8.1.4. Port of Rotterdam

Ports provide several functions to shippings lines that are crucial for the efficiency of these shipping lines. The main function is to provide a safe location where vessels can be loaded and offloaded. Ports are responsible to improve shore-based facilities. For example when vessel become bigger and need deeper water, the port is responsible to provide approach channels and berths the vessels can use. Port also must provide enough space to handle and store all different kinds of cargo. Last the port is responsible to connect the hinterland transportation methods efficiently to the port and make sure they are integrated into the port operations[58].

As already mentioned, the market is very competitive and ports compete on the number of vessels they handle and the amount of cargo[58]. Ports compete heavily for vessels to (off)load at their location. New concepts are able to change the landscape of power between ports. Ports that are leaders now, may be out of business in the near future if they do not adjust. So the Port of Rotterdam needs to keep innovating to provide relative high-level services, low waiting and handling times and a value adding hinterland transportation network[58].

8.2. Added value

After identifying and discussing the relevant stakeholders, the added value of the proposed frameworks and algorithms is discussed in this section for every stakeholder.

8.2.1. Shipping lines

Shipping lines are motivated to deliver the containers on time, if they are late a penalty needs to be payed. Furthermore as noted in the previous section fuel cost are reduced by traveling as slow as possible. These two statements contradict each other, on the one hand shipping lines want to travel as fast as possible for timely delivery but on the other hand as slow as possible to save fuel. This contradiction leads to a specific kind of behavior, vessels first travel at maximum speed and when they are certain the deadline will be met they start traveling as slow as possible. So an ETA prediction tool can have value for a shipping line in three areas[58]:

1. The tool is able to give the shipping line a more accurate prediction of their arrival time, so they have more certainty if they can meet their deadline. So the prediction can indicate if a vessel needs to speed up to meet the deadline or can slow down to save fuel.
2. If the tool was able to predict the speed a vessel needs to travel to meet their deadline. In this case the variations in speed can be minimized which even further reduces the fuel consumption (cost).
3. The tool can also be used as a competition monitoring tool. If the tool provides information on how many vessels are headed to a port, the available supply for a new shipment to that port can be known. This can be used in price negotiations by the shipping lines.

8.2.2. Terminal operators

An ETA prediction tool provides the biggest added value to the terminal operators[58], because accurate ETA information is vital for the planning of the operating activities of a container terminal[44]. The tool is most effective for terminal planning when it is able to provide an ETA 2 to 3 days before the arrival of a vessel[58]. The tool is able to add value in a few ways.

The uncertainty over the arrival time of vessels is reduced by more accurate predictions, so the planning of the berths can be improved. The berths are allocated based on the expected time of arrival of the vessels, so when a vessel is late, there may not be enough space available at the quay which lead to waiting times. But this does not only impact the vessel that is late, but also vessels arriving after that vessel since the whole planning needs to be adjusted. Another problem occurs when a vessel arrives before schedule, because when a vessel arrives early it may have to wait until its allocated time slot. But a vessel may also be serviced before schedule, when this happens the containers that needs to be loaded onto the vessel may not be available for transport. For example, other containers that were scheduled to be loaded on another vessel may still be on top of the needed containers[58].

Changes in the berth allocations, due to the uncertainty of vessel arrivals, also negatively influence the yard planning schedule. Vessels may be assigned to another berth, therefore the cargo that needs to be loaded needs to be moved from its initial position to the new berth. This leads to extra movements and thus an increased workload and waiting times[58]. Also the allocation of the cargo that needs to be offloaded needs to change to be stored near the berth or the cargo needs to be transported across the yard to the initial allocated position which also increases the amounts of movements. Heij[22] showed that unscheduled arrivals may lead to a peak in energy consumption which increases the energy consumption for the terminal and thus the annual cost because energy consumption has a large impact. The energy increases because of the extra movements, so less unexpected arrivals means less movements and thus less energy consumption costs.

To cope with the uncertainty and the resulting workload increase, more resources are assigned for every work shift then necessary, which leads to an increase of the costs[13]. So an ETA prediction tool can give more certainty about the arrival time of a container, which lowers number of container movements, reduces disruptions in the planning and thus reduces the workload. With the reduced workload less resources are needed and therefore the operational costs are reduced[58].

8.2.3. Forwarders and hinterland carriers

A vital piece of information for the planning of a forwarder is the ETA of incoming container vessels, because the arrival of a vessel is the starting point of all the activities in the supply chain related to hinterland container transport[58]. The research of Menger[44] showed that information regarding the delays and estimated arrival times of container vessels is very high on the list of demands regarding information hinterland carriers would like to have. More accurate ETA times allows the forwarders to book the necessary capacity with the hinterland carriers without misjudging the demand[58].

The current booking process of forwarders consists of two steps[58]:

1. A week ahead a fixed capacity is booked with the rail and barge hinterland carriers. If the forwarder is not able to book capacity at a train or barge, the container is transported by truck. When demand is misjudged this leads to an increase in cost. When demand is judged to be lower more ad-hoc trucks needs to be booked which are expensive and thus the cost increase. When demand is judged to be higher, booked capacity remains unutilized which lead to an increase in average costs per container.
2. When a container actually arrives the container is assigned to a hinterland transportation mode with available capacity based on the preferences of the shipper.

Furthermore terminals give priority to deep sea vessels over barges. So when a vessel arrives early or late it may occupy a berth assigned to a barge. This leads to extra cost for the hinterland carrier because the barge needs to wait and may force the forwarder to ship a container via truck to ensure timely delivery. If a tool is available that predicts the arrival of a deep sea vessel, forwarders could identify disruptions in advance and change their planning. In this way they can avoid double cost for booking capacity on both barge and truck for example[12].

So it is clear from the information above that the forwarder needs a prediction horizon of a minimal of 7 days. When the time horizon is at least seven days, forwarders can better estimate the demand for capacity. When the preferences of the shipper are also coupled with this prediction a forwarder is even able to better predict the needed capacity for every transportation mode. When more certainty about the ETA of a vessel is at hand, more containers can be booked with barge and train carriers. This not only reduces the transportation cost but also reduces the impact on the environment[58].

8.2.4. Port of Rotterdam

The Port of Rotterdam is the leading port in Europe in this highly competitive market. To consolidate its position the Port of Rotterdam needs to keep innovating. So a differentiation strategy may help the Port of Rotterdam to sustain its competitive advantage. An ETA prediction tool can play a crucial role in this strategy[58]. The Port of Rotterdam states in his annual report of 2014 that it wants to be the leader on the area of innovation development and innovation use. Furthermore they highlight that they want to further improve the efficiency of the transportation network and develop data and data application[21]. An ETA tool perfectly fits in this strategy of the Port of Rotterdam.

If the Port of Rotterdam enacts as the enabler of an ETA tool and provides the tool to the stakeholders in the hinterland transportation supply chain as identified in this section, these stakeholders are able to improve their performances and as a result the Port of Rotterdam attracts more vessels to their port. So the tool gives the Port of Rotterdam a competitive advantage. Next to the benefits for the other stakeholders, an ETA tool also has benefits for the planning of the Port of Rotterdam itself. For example, the Port of Rotterdam is responsible for assigning tugboats to guide a vessel to the terminal. With an improved ETA the Port of Rotterdam is better able to plan these activities and improve the availability of tugboat pilots. It can also reduce traffic around the port area because less trucks are needed and trucks are able to arrive closer to the actual release time of a container[58].

8.3. Power to improve data quality of AIS

To reap the full potential benefits of the proposed frameworks and algorithm, changes need to be made to the AIS messages sent by vessel in order to improve the data quality. But are the stakeholders able to improve the data quality of AIS messages? The ability of stakeholders to change AIS is discussed in this section.

8.3.1. Shipping lines

The personnel of a shipping line aboard a vessel is responsible for inputting the voyage related information of AIS messages. So a shipping line can influence the data inputted by company regulations. This may improve the data quality of AIS and reduce false information that is provided maliciously. However since multiple shipping lines operate in the domain of containerized maritime transport, they may not impose the same regulations regarding the data input for AIS messages and therefore the resulting messages can still differ. Although it will make the data processing step easier, it will not completely solve the problems with AIS messages.

8.3.2. Terminal operators

Terminal operators are only able to use AIS messages if they have their own receiver or if the AIS messages are provided to them by the port. Since terminal operators are on the receiving side and have minimal direct contact with shipping lines, they are not able to influence the data quality of AIS messages although the terminal operator would have the biggest benefits.

8.3.3. Forwarders and hinterland carriers

Forwarders and hinterland carriers are in the same situation as terminal operators regarding AIS messages. They only receive AIS messages when they have their own receiver or when AIS messages are provided by the port. Just like the terminal operator they are on the receiving side and have no direct contact with the providers of AIS messages and therefore can not influence the data quality of these AIS messages.

8.3.4. Port of Rotterdam

Ports are one of the possible recipients of AIS messages. The Vessel Traffic Services(VTS) of a port use the AIS data to guide the vessels through traffic and other obstacles to their berth in the port and use AIS to communicate with vessels. Ports are able to provide this AIS data to other stakeholders related to the port, for example the terminal operators and hinterland transportation parties, or provide only relevant information to these stakeholders. Ports are not able to directly influence the data quality of AIS messages, since they are the receiving party. However a port can try to influence the data quality by imposing fines on false or wrong information. But due to the competitive nature of the market the risk exists that shipping lines will visit other ports that do not enforce these fines. Therefore ports are not able to influence the data quality of AIS messages.

8.3.5. International Maritime Organization

Since the stakeholders in the supply chain have little to no influence on the quality of AIS messages, stakeholders outside the supply chain that can influence the data quality are identified. The only stakeholder that can influence AIS is the International Maritime Organization.

The International Maritime Organization (IMO) is the global standard-setting authority of the United Nations. The main task of the IMO is to create a fair playing field for all stakeholders in the maritime domain by imposing legislation and standards[27]. In this capacity the IMO is the owner of the AIS standard and is therefore able to enforce changes in the standard. For instance they are able to change the manual unrestricted input of destinations into a restricted input that only uses Port Codes. However the IMO is not part of the supply chain so it will not be impacted by the resulting benefits directly, but they will incur cost for developing and training for instance. But since the IMO is engaged in creating an institutional framework necessary for a green and sustainable global maritime transportation system[27], they may feel inclined to change the standard when it results in a more green and sustainable global maritime transportation system. As shown in the previous section the ETA tool may decrease the cost in the supply chain, thus increasing the sustainability of the transportation system. And also reduce the fuel consumption of container vessels and energy consumption in terminals. This has a positive influence on the environment making the transportation system more green, thus achieving one of the goals of the IMO.

8.4. Implementing changes to AIS

So the power to change AIS does not lie with the stakeholders that reap the benefits of these changes. Making changes to AIS comes with costs for development and implementation and at least the cost for development are costs for the IMO. This could be problematic since the IMO might not feel the need to implement changes and to bear the cost for development. A possibility to break this standstil could be for a group of shipping lines, port or terminal operators to pay for the changes. But since the market is highly competitive, these stakeholders are not able to increase their prices to cover the costs. By increasing the price these stakeholders could lose clients that choose for cheaper options that also reap the benefits without paying. So it is improbable that a group of the stakeholders in the supply chain will pay for these developments, since they can not earn back the costs. The most probable option is constructing a consortium with the top 20 ports, since these ports can not be neglected[63] they are able to increase their prices and earn back the costs.

However the changes have benefits that align with the mission of the IMO. So when the IMO wants to implement the changes but does not want to bear the costs, they might impose a license fee on the use of AIS. And since AIS is mandatory the cost are equally divided over all the stakeholders and thus have no impact on the market position of these stakeholders. However the question how to implement the changes is a question for a research in itself and lies outside the scope of this research. Some possiblities are briefly discussed but further research is needed to cover all posibilities and come up with a strategy.

8.5. Conclusion

The research question: *What is the added value of the proposed algorithms and framework for stakeholders in the supply chain and how can stakeholders influence the data quality of AIS messages?* is answered in this chapter.

The different stakeholders in the supply chain reap different benefits from accurate ETA predictions. The Port of Rotterdam is able to improve its competitive position in the market for instance, terminal operators are better able to make a planning for their resources and hinterland parties are able to plan the hinterland transportation more accurate. Also the shipping lines benefit from more accurate predictions because they are able to save on fuel and make an assesment of their competitive position.

However these stakeholders, except for the shipping lines, are not able to influence the data quality of AIS messages. Shipping lines can impose company regulations but this may not result in a new standard and also does not solve all problems with the data quality. The only stakeholder that is able to make changes to AIS and thus increase the data quality is the IMO. At first we would not expect the IMO to make changes to the standard because of the resulting costs for them without reaping the benefits. However since implementing an improved standard might result in a more sustainable and green global maritime transportation system, the IMO might be inclined to implement an improved standard since this is their main goal. However making changes to the AIS standard will come with costs. The IMO might not want to incur these cost and therefore changing AIS might become a problematic process. How to overcome this process and designing a strategy needs further research and lies outside the scope of this research.

9

Insights and suggestions

The proposed frameworks and algorithms are able to improve the data quality of AIS messages and make predictions based on AIS and routes. But results are generated that can be improved upon. In this chapter the insights that are obtained during the research are discussed and suggestions are made to improve the data quality of AIS messages and the predictions of the framework.

9.1. Data quality

One of the most important aspects to make improvements possible in a supply chain is reliable data capture, storage, processing and communication[36]. Therefore the influence of the data quality on the research is discussed and suggestions are presented how to improve the data quality by changing the AIS standard. First general data quality problems are discussed, followed by a discussion of problems that are specific to this dataset.

9.1.1. Quality of AIS data

The biggest problem with AIS data is that vessels are completely unrestricted in how to input their destination. Because no rules are being imposed a lot of different notations exist. Some vessels use port codes, while others use names. Some vessels state also the country, while other vessels state both their port of origin and port of destination in the destination field. Also vessels provide terminal specific information. So using the destination field for route classification is very difficult. All the different notation need to be standardized and then translated into a standard name for each port. Two websites are scraped to construct a database of port names and port codes. But the problems with the created port database is that it contains every port worldwide and thus also ports that have no container terminals. So when the name of a port with a container terminal and the name of a port without a container terminal are very similar and the vessel made a typo in their destination field, the standardization algorithm may standardize to the wrong port or is not able to choose between the ports. This problem is circumnavigated by using the retrieved WPI database with location information. But since also this data base contains a lot of irrelevant ports, for instance the marina of Maassluis, doing standardization based on location information with these ports is also error prone. The marina of Maassluis is situated across ”De Nieuwe Waterweg” of some terminals of the Port of Rotterdam. Since this marina is closer then the location information of the Port of Rotterdam, the marina of Maassluis is given as the destination, which is obviously wrong.

To alleviate these problems two possible solutions are proposed. The first solution would be creating a database that contains the port names, port codes and location information, where a geographical region is defined. Based on the location information of a vessel, routes are created in combination with this database. A geographical region is used instead of a point because a port visit can be defined as a stop in that geographical region. The second solution is to create a database with all the errors encountered in the destination field and a mapping to a correct name for a port. This database can be created after the first run of the proposed framework. The first solution is preferred because this is more flexible and is not depending on a previous occurrence of an error. Also the first solution can be used for another problem with the methodology which are discussed later.

Another problem when using AIS data is the navigational status. Also this part of the AIS message is manually set and is sometimes not changed when the navigational status of a vessel actually changes. Because the framework is dependent on the changing of these status to discover arrivals in a port, the navigational status is checked beforehand. By changing the status messages based on a port database that contained a lot of ports, that are also irrelevant, it is possible that statuses are changed when they should have not been changed. This has a negative effect on the performance of the algorithm and can cause special cases.

The draught of a vessel may also cause problems. As concluded from the literature review it is likely that the wrong draught or no draught is stated by a vessel. This may have a negative impact on the accuracy of our predictions. However since a historical dataset is used, the provided information must be relied upon. In the dataset all vessels reported a draught so no manipulation is needed.

9.1.2. Quality of the dataset

Now general problems with AIS are discussed, problems that are related to the dataset are discussed. As already noted in Chapter 7, the origin of the dataset is somewhat unclear. The dataset has been provided by TNO but it is unclear who created the dataset. Due to this uncertainty, the origin of the information regarding the ETA of a vessel is also uncertain. This could be either from AIS messages or from communication with the shippers agent. Either way the information about the ETA of a vessel consists a lot of errors. In a lot of cases the ETA was not changed and remained the same for multiple voyages. This could have a few causes, either the ETA was really not changed, something went wrong while the AIS messages were decoded or ETA's were wrongly associated while creating the dataset. Therefore we needed to remove a lot of voyages since it is very hard, maybe even impossible, to make predictions regarding the estimates of a captain. Furthermore some voyages remained in the AIS dataset with a very large estimation error for the Hamburg and Felixstowe route. These voyages hampered the performance of the prediction models for these routes.

The dataset that was provided by TNO has already been sampled, so only one observation per hour remained per vessel. Although this significantly reduced the size of the dataset, it also was the cause of some challenges in the dataset. Because of the sample rate it was possible that between two observations the destination of a vessel had already been changed while moored. If the first observation was before mooring and the second while moored, port visits are missed, because the framework depends on the status of a vessel and the destination as stated in the AIS message. If a database is created with location information regarding ports as proposed in the previous section this problem is solved because port visits are identified based on the geographical information in the AIS message.

Another problem with destinations has been observed in the AIS dataset. In some cases the destinations have not been changed by a captain and therefore ports visits are missed and thus wrong routes are created or a wrong set of possible routes based on a wrong voyage is identified. Also this problem can be solved by creating the proposed database.

In general the poor data quality of the dataset had a large influence on the research. First of all due to the poor quality a lot of voyages are removed. Therefore the AIS dataset with correct information was very small and this had a negative influence on the amount of routes in the dataset and the performance of the predictions. One of the basis assumptions was that vessels travel using repetitive patterns. However because the data is cleaned so rigorously, these repetitive patters remained hardly present in the dataset and therefore very few voyages per route were observed. Therefore the amount of observations to make the predictions on is low and thus the predictions are not very accurate.

Second, due to the poor quality of the data, a lot of manipulation is performed in the AIS dataset. This took a considerable amount of time, both regarding the research and regarding computation time. A lot of the research time is invested into the manipulation of the dataset and therefore some functionality is not added to the prediction algorithm that would have improved the predictions. Also manipulating the dataset takes a lot of computation time, which is not desirable. Therefore in Section 9.3 a discussion is presented on how the predictions could be improved and also how to speed up the entire program.

9.2. Changes to AIS

AIS is still very error prone and therefore changes to the AIS standard are suggested. When these changes are incorporated, AIS data will be less error prone and more accurate. As discussed in the stakeholder analysis, the power to change AIS lies with the IMO. However they are not part of the supply chain so they can not reap the benefits directly. However the indirect benefits of the proposed ETA predictions, that may be increased by the proposed changes, are aligned with the mission of the IMO. Therefore the IMO may feel inclined to implement the changes.

The first proposed change is using the instrument aboard a vessel more extensively to do crosschecking or even fully rely on these instruments. The draught of a vessel is read from a vessel and manually inputted into the AIS message, it is proposed that the draught is automatically read from the instrument and inputted into the AIS message. Also the navigational status of a vessel is manually inputted, but for instance the speed of a vessel is known. So the speed of the vessel can be used to crosscheck the navigational status. If each status has certain restrictions when that status can be used, this may increase the data quality of the navigational status, enforcing the correct status in the AIS messages.

The second proposed change is how to set the destination of a vessel. Currently 20 characters are available to put in anything deemed necessary. This really deteriorate the data quality of AIS messages. Therefore it is suggested to change this into a more restricted input. When inputting the destination one is able to search for ports from a database. Then one can select the port that is the destination and the Port Code of this port is automatically inputted into the AIS message. By enforcing this standard input the data quality of the destination variable will increase substantially and the variable will be much easier to use in research.

When a destination is changed this means that in almost every case the ETA will also change. Therefore it is proposed that a change in the ETA is enforced when changing the destination. By enforcing the ETA to be changed when a destination is changed, each ETA is related to the destination that is stated in the AIS message. Furthermore a suggestion is made that error messages will be displayed when an ETA has passed, enforcing new ETA's. This can even be expanded to error messages just before the timestamp of the ETA so the vessel needs to confirm the ETA or can change the ETA when needed. This increases the accuracy of the ETA.

9.3. Improvements to the framework

The current prediction algorithm does not take a lot into consideration and is very basic. Some ideas to improve the predictions are not incorporated into the algorithm due to time limitations. Therefore these suggestions are discussed in this section, so they can be incorporated in future research.

First of all very common occurrences in maritime shipping have not been taken into account. Vessels could be laying at anchor in an anchorage, which means that a vessel already arrived but needed to wait until a berth was available. So basically a vessel already arrived at a port but could not go to a berth. So the ATA at a port is actually earlier than identified by the program. Also the time that a vessel spends at these anchorages has an influence on the travel times, so this should be taken into consideration. Next to waiting in anchorages, it could also be possible that the vessel needs to wait at a lay by berth in the port, this also has an influence on travel times and the time spent in a port and should therefore been taken into consideration.

Vessels that travel to or from for instance Russia can take two routes. They can travel around Denmark or travel via the Kiel Canal, this has a considerable influence on the travel time of a vessel. Since most of the vessels state when they travel via the Kiel Canal this can be taken into consideration. However the researchers had not enough knowledge of the domain and were not aware of the Kiel Canal. Therefore the information regarding the Kiel Canal have been removed from the dataset and thus could not be used in the predictions. As an improvement is suggested that an extra boolean attribute is introduced to the dataset that is set to true if the vessel travels via the Kiel Canal.

Since the time vessels spend in ports or anchorages is highly variable and greatly influences the travel times of a vessel, these times need to be incorporated in the framework. A suggestion is that the average waiting times are calculated based on characteristics of the vessels. So vessels can be classified based on for instance size and shipping line and the influence of this suggestion should be researched.

More generally speaking, in this research not a lot of effort have been put in finding the optimal set of predictors or testing different machine learning techniques or configurations. When this is further researched upon, the performance of the framework might increase. A possible predictor to include extra, next to the aforementioned predictors, is time of year. For instance the Baltic Sea is more difficult to travel in winter and this may impact travel times. Furthermore research could also be conducted into tailoring prediction models to specific routes, so one route is predicted by KNN-regression and another by a Neural Network. But also setting K for KNN-regression per route.

In the research of Lane et al. they propose a method in section 2.1 that can be used to identify the next destination of a vessel[35]. This method should be incorporated in the method so that the destination that is stated by a vessel can be checked. By doing this, wrongly provided information could be identified in an early stage.

Prediction models have now been created based on entire routes, it might be possible that routes are not travelled as repetitive as is assumed. Therefore it could be that an entire route is sparsely travelled, although the different sub-voyages between ports are heavily travelled. Therefore predictions should be made based on these sub-voyages in combination with predictions regarding waiting times in ports and anchorages. In this way it might be possible to make more accurate predictions based on more data points.

Another case that is not incorporated in the algorithm are changing destinations. It is possible that a vessel changes its destination while at sea. Currently this is not observed by the algorithm and the next destination is added to the sequential representation like both ports were visited. By constructing a database as was proposed in the previous section, this problem can be solved. Then routes can be constructed by identifying actual port visits and not by using the destination information as stated in the AIS messages. When the database is not created, destinations should be added only when a vessel is moored to the sequential representation. And for route identification a temporary combination of the sequential representation and destination can be used. This will result in more accurate routes.

Furthermore new previously unseen routes are not added to the list of routes. Most likely is every possible route covered in historic datasets. But since it is possible that new terminals are created this would be a function that would be nice to have in the algorithm.

Currently a set of possible routes is identified for each vessel based on its current voyage and IMO number (when possible). This means that it is possible for a single observation to have multiple predictions. Since the current methodology is not able to select a single route when multiple routes are available, all possible predictions are presented with additional information, the vessel and route used for specific prediction, to the user of the framework. It is then up to the user to decide which prediction to use in the planning. Selecting a single route should be incorporated into the methodology, but unfortunately due to time constraints is not. So a subject for further research as one of the possible improvements to the methodology is improving the route identification.

Since the dataset is cleaned to vessels visiting the Port of Rotterdam and a travel time threshold is set, an area under observation was not set and vessels that pass an area under observation but do not visit Rotterdam did not need to be handled. However in practice the actual travel time is unknown and it is possible that vessel pass the area under observation but do not visit Rotterdam. Therefore the area under observation needs to be included and an extra check needs to be included that if the vessel travels outside the area that is under observation, the vessel is removed from the list of observed vessels.

The last case that should be improved is the calculation of the estimated travel times. If a vessel correctly sends its ETA this changes for every destination. However the dataset shows that the ETA is hardly changed in a correct manner and therefore it is hard to do the calculations of the estimated travel times. Currently the ETA is used that is sent when the vessel is moored for the very first time in the Port of Rotterdam and thus is the best guess of the ETA by the vessel. However it could be that earlier in the voyage a completely other ETA was communicated, but because knowledge about which ETA is related to which port is lacking, ETA's stated in an earlier part of the voyage are not utilized.

9.4. Speeding up

One of the problems in the research was the speed of the framework. This had a few reasons. First of all, based on the experience with R of the researchers, R is selected as the programming language. But R is by default run on a single core and therefore not the quickest option available. However this limitation was not known to the researchers at the start of the research and when this knowledge was gathered, not enough time was left to change to another programming language. Efforts have been made to run the algorithm with R in parallel, but these have not been successful since it requires a lot of special solutions that are not always obvious or would need changes to the entire program. Another reason for the slow calculations was the hardware that was used in the research. Since access to a High Performance Computer (HPC) was provided at a very late stage, the program has been run on a laptop, although this laptop packs a lot of computing power the calculations still took considerable time, about a week. With a quicker computer this time could be diminished.

A possible solution to speed up the algorithm could be multithreading. Since the vessels are all independent from each other, the dataset can be multithreaded based on the IMO number of a vessel in the first step. After assigning an ID to every voyage multithreading can be done based on the IMO number and voyage because voyages are also not related to each other. By introducing multithreading to the program huge speed ups can be achieved.

Other solutions are rewriting the algorithms in another programming language that has more support for parallel programming. Adjusting the algorithms written in R to do parallel programming is another suggestion that may speed up the framework.

9.5. General remarks

In general although AIS data is considered as a reliable information source in the maritime domain in a lot of literature, it still has a lot of problems and is therefore less usable than expected. The quality of the data has a big influence on the usability of AIS data and it takes a lot of time and effort to clean and manipulate the data so it can be used. Especially data that needs to be inputted manually can hardly be relied upon. Therefore it is advised to either focus on data that is not inputted manually, like the location information, and use this information in combination with a database that contains the geographical region of a port. Or to stay away from using AIS data until the proposed changes have been implemented by the IMO, these changes will greatly enhance the data quality of AIS messages.

10 Conclusion

The sub research questions as presented in Chapter 3 are answered in the thesis. In this chapter these answers are summarized and combined to answer the main research question. At the end of the chapter areas for further research as discussed in Chapter 9 are presented.

10.1. Data quality

The first question that is answered is *What are possible issues with the data quality of AIS messages?* In Chapter 5 issues with variables of AIS messages and AIS messages in general are identified. Variables in AIS messages are error prone, especially variables that are manually inputted. These variables may show intentionally wrong information or may not be updated. Another possibility is that no information is represented in these variables. Changes to improve the data quality of AIS messages are proposed in Chapter 9.

Next to the problems with the manually inputted variables, the completeness of the data is also a problem. During processing multiple voyages with large gap or that were poorly described by the AIS messages were encountered. This problem can be solved by interpolating the missing messages, but has not been done due to time constraints.

10.2. Pre-processing AIS data

In Chapter 5 the question *How can AIS messages be pre-processed to improve the data quality so it can be used for route identification and ETA prediction?* is answered. As is shown with the first research question, AIS data has some issues. Therefore the AIS data first needs to be processed before it can be used. Especially variables that need to be inputted manually cause a lot of problems. In Chapter 5 some variables are identified that should be checked and translated this to steps that should be taken while pre-processing AIS data in order to improve the data quality of AIS messages and be able to perform route classification and predict the ETA of a container vessel. Furthermore steps are included to construct variables that are needed while predicting the ETA of a container vessel. A workflow of these steps is shown in Figure 10.1. These workflow shows general steps that need to be taken, the steps need to be adjusted to the dataset and it may be possible that some additional steps are required based on the dataset. However with the current AIS standard all these steps are necessary to create a dataset of a sufficient quality to do route identification and ETA predictions.

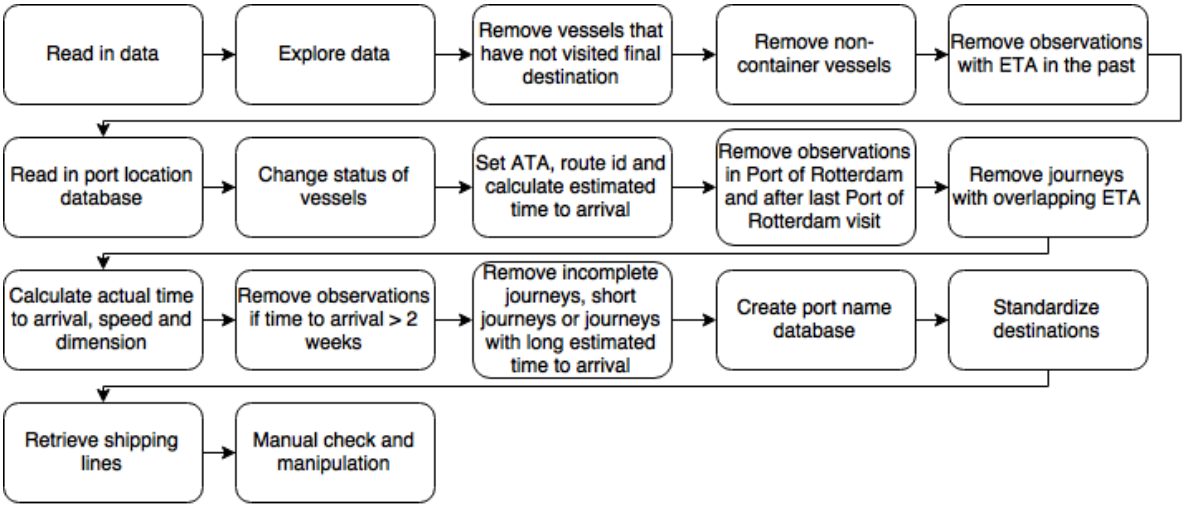


Figure 10.1: Workflow to improve the quality of an AIS dataset.

10.3. Route identification

Once the AIS data has been processed it can be used to identify a set of possible routes for a vessel. But *How can a set of possible routes of a container vessel be identified using pre-processed AIS data?*. Insight from the research of Pallotta et al.[57] and Lane et al.[35] are used to create a simple method to identify a set of possible routes. The steps are presented in Figure 10.2. For 99.3% of the AIS messages the correct route is represented in the set of possible routes.

10.4. Combining route identification and ETA prediction

In this chapter the research question *How can the ETA of a vessel be predicted with the use of pre-processed AIS data and route identification?* is answered. Routes can be used to select prediction models that are based on specific routes, however it is unclear if this improves the performance of the predictions and further research is needed. But in general using machine learning makes it possible to make prediction that are of equal quality in comparison to the best guess of the vessel's crew. Algorithm 10.1 shows a framework to do model selection based on routes and predict the ETA of a vessel based on a pre-processed AIS message. The ETA as communicated via the AIS message is identified as an important predictor in most cases. However in some cases this hampered the predictions. So further research is warranted that assesses which predictors to use for which routes.

Algorithm 10.1 Framework for ETA prediction

Input: AIS database with standardized destinations, database with possible routes and vessels that travelled those routes

1: Create training set for each route and test set from AIS database
2: Create prediction model for each route
3: Read AIS message from test set
4: Identify possible set of routes for vessel *# Algorithm 6.3*
5: Select prediction models based on possible set of routes
6: Do predictions

Output: Prediction for each possible route

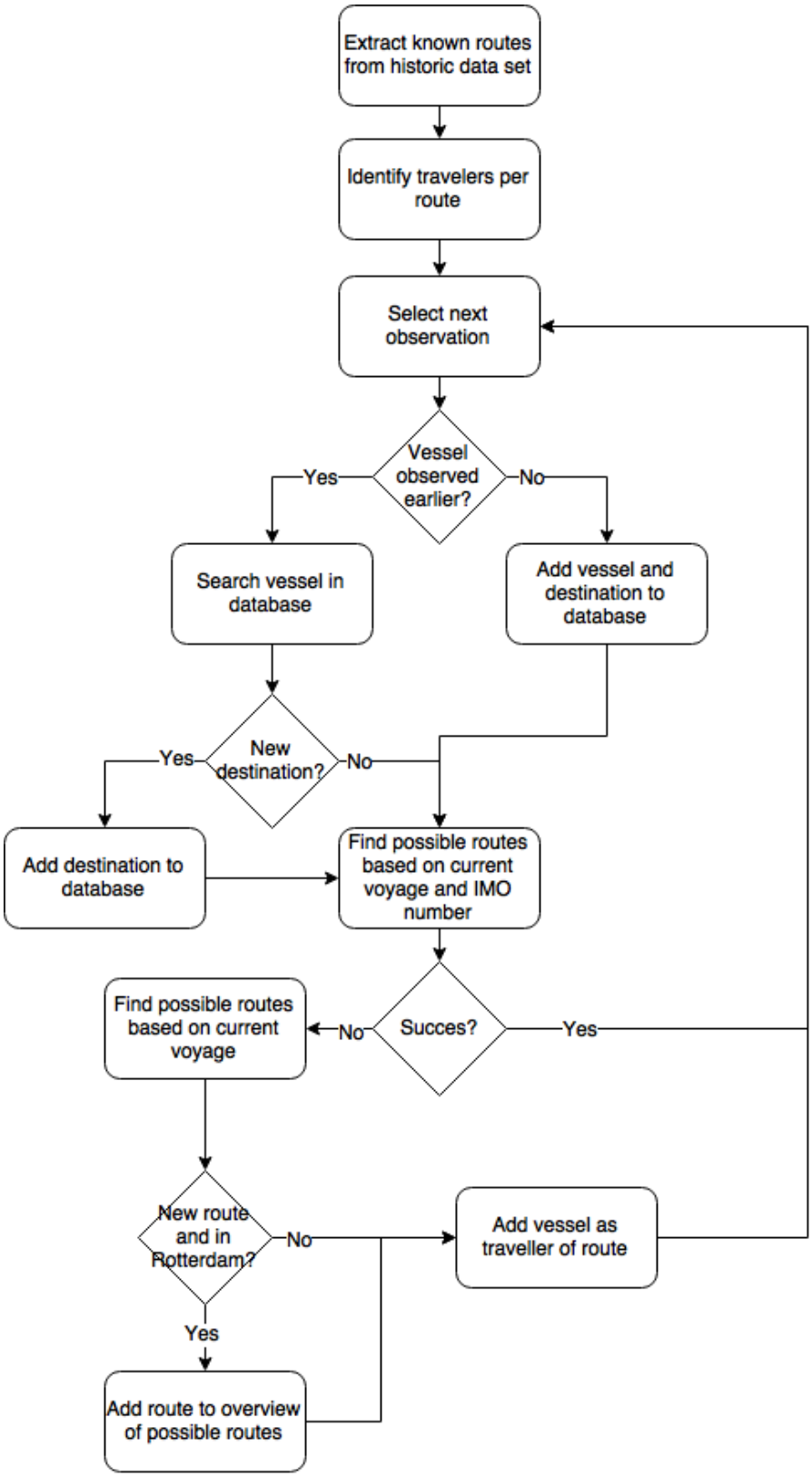


Figure 10.2: Route identification framework

10.5. Added value and changing AIS

To answer the question *What is the added value of the proposed algorithms and framework for stakeholders in the supply chain and how can stakeholders influence the data quality of AIS messages?*, a stakeholder analysis is made in Chapter 8. All stakeholders in the supply chain reap benefits from more accurate ETA predictions but do not have the power to change the AIS standard or to a small degree. The stakeholder that is able to make changes to AIS is the IMO, but the IMO is outside the supply chain so will not directly reap benefits from changing the AIS standard but do incur some of the cost. Therefore changing the standard might not be an obvious option, the IMO has a mission statement and the indirect benefits of changing the AIS standard align with this mission. So the IMO may feel a need to change the standard. Changing a standard will come with costs and the IMO might not want to cover these costs. So changing AIS to improve the data quality may not be easy and a strategy needs to be designed, but this is subject to further research.

10.6. Final conclusions

To answer the main research question: *How to improve the AIS-based ETA predictions of vessel en route to a port by leveraging route identification?* the answer to the questions as discussed above are combined. Most important to improving AIS-based ETA predictions is the data quality of AIS messages. In Chapter 5 data quality issues are assesed and a framework is designed to improve the data quality. This framework is shown in Figure 10.1. In Chapter 6 insight are taken from two route prediction methodologies to construct a route identification framework. This framework, as shown in Figure 10.2, is able to identify a possible set of routes based on the current voyage of a vessel and its IMO-number. To leverage this route identification, Algorithm 10.1 is presented in Chapter 7. The performance of the prediction algorithms is equal to the best guess of a vessel's crew, but the effects of incorporating routes are unclear and need further research. The benefits of improved predictions are positively impacting the stakeholders inside the supply chain. The benefits are enlarged when the data quality of AIS messages is improved, but these stakeholders are lacking the power to do so. The IMO is needed to make changes to the AIS standard and thus bear the cost, but the IMO is not affected by the benefits. Therefore a strategy needs to be designed and implemented to change the AIS standard, this is subject to further research. This is only one of the insights and suggestions as discussed in Chapter 9, further improvements to the different frameworks are possible and presented in the next section.

10.7. Future research

In Chapter 9 some suggestions for further research were introduced. The suggestions are listed below. For an explanation we refer to Chapter 9.

- Improve algorithms. For instance incorporate waiting times in port into predictions per port and vessel (category) or incorporate Kiel Canal
- Develop a strategy to implement the suggested changes to AIS
- Find the optimal set of predictors, best prediction method and configuration per route
- Incorporate next destination identification by Lane et al.
- Predictions based on sub-routes
- Route classification

Bibliography

[1] Karl Gunnar Aarsæther and Torgeir Moan. Estimating Navigation Patterns from AIS. *Journal of Navigation*, 62(04):587, October 2009. ISSN 1469-7785. doi: 10.1017/s0373463309990129. URL http://dx.doi.org/10.1017/S0373463309990129.

[2] N. Adrienko and G. Adrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219, Feb 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2010.44.

[3] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statist. Surv.*, 4:40–79, 2010. doi: 10.1214/09-SS054. URL http://dx.doi.org/10.1214/09-SS054.

[4] Nicholas J Bailey. Training, technology and AIS: looking beyond the box. In *Proceedings of the Seafarers International Research Centre's Fourth International Symposium*. Seafarers International Research Centre (SIRC), 2005.

[5] Michael Baldauf, Knud Benedict, and Florian Motz. Aspects of technical reliability of navigation systems and human element in case of collision avoidance. In *Proceedings of the Navigation Conference and Exhibition, London, UK*, volume 2830, page 111, 2008.

[6] Neil Bomberger, Bradley Rhodes, Michael Seibert, and Allen Waxman. Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness. In *2006 9th International Conference on Information Fusion*, pages 1–8. Institute of Electrical and Electronics Engineers (IEEE), July 2006. doi: 10.1109/icif.2006.301661.

[7] Leonid Boytsov. Indexing methods for approximate dictionary searching: Comparative analysis. *J. Exp. Algorithmics*, 16:1.1:1.1–1.1:1.91, May 2011. ISSN 1084-6654. doi: 10.1145/1963190.1963191. URL http://doi.acm.org/10.1145/1963190.1963191.

[8] S.J. Chang. Development and analysis of AIS applications as an efficient tool for vessel traffic service. In *Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600)*, volume 4, pages 2249–2253. Institute of Electrical and Electronics Engineers (IEEE), November 2004. doi: 10.1109/oceans.2004.1406499.

[9] T De Boer. An analysis of vessel behaviour based on AIS data. Master's thesis, Delft University of Technology, 2010. URL http://repository.tudelft.nl/islandora/object/uuid%3A25255610-e276-417a-bc2d-a3916c07e348?collection=education.

[10] I. De Vreede. Managing Historic Automatic Identification System data by using a proper Database Management System structure. Master's thesis, Delft University of Technology, 2016.

[11] Urška Demšar and Kirsi Virrantaus. Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science*, 24(10):1527–1542, 2010. ISSN 1365-8816 1362-3087. doi: 10.1080/13658816.2010.511223. URL http://dx.doi.org/10.1080/13658816.2010.511223.

[12] Alexander Dobrkovic, Maria-Eugenia Iacob, Jos van Hillegersberg, Martin R. K. Mes, and Maurice Glandrup. Towards an Approach for Long Term AIS-Based Prediction of Vessel Arrival Times. *Lecture Notes in Logistics*, pages 281–294, August 2015. ISSN 2194-8925. doi: 10.1007/978-3-319-22288-2_16. URL http://dx.doi.org/10.1007/978-3-319-22288-2_16.

[13] Gianfranco Fancello, Claudia Pani, Marco Pisano, Patrizia Serra, Paola Zuddas, and Paolo Fadda. Prediction of arrival times and human resources allocation for container terminal. *Maritime Economics & Logistics*, 13(2):142–173, June 2011. ISSN 1479-294X. doi: 10.1057/mel.2011.3. URL http://dx.doi.org/10.1057/mel.2011.3.

[14] Finland. Agenda item 5: Matters arising from other helcom meetings, 2007. URL https://portal.helcom.fi/Archive/Shared%20Documents/AIS%20EWG%2016-2007_5-2%20Quality%20of%20AIS%20information.pdf.

[15] Rebeca Gómez, Alberto Camarero, and Rafael Molina. Development of a Vessel-Performance Forecasting System: Methodological Framework and Case Study. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 142(2):04015016, 2015.

[16] Allan Graveson. AIS – An Inexact Science. *Journal of Navigation*, 57(3):339–343, September 2004. doi: 10.1017/s0373463304002759. URL https://www.cambridge.org/core/article/div-class-title-ais-an-inexact-science-div/35A48D75701F33E79F1B44EECA10E50D.

[17] Marco Guerriero, Stefano Coraluppi, Craig Carthel, and Peter Willett. Analysis of AIS Intermittency and Vessel Characterization using a Hidden Markov Model. In *GI Jahrestagung (2)*, 2010.

[18] Hans-Otto Günther and Kap-Hwan Kim. Container terminals and terminal operations. *OR Spectrum*, 28(4):437–445, July 2006. ISSN 1436-6304. doi: 10.1007/s00291-006-0059-y. URL http://dx.doi.org/10.1007/s00291-006-0059-y.

[19] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier LTD, Oxford, 2011. ISBN 0123814790. URL http://www.ebook.de/de/product/14641128/jiawei_han_micheline_kamber_jian_pei_data_mining_concepts_and_techniques.html.

[20] Abbas Harati-Mokhtari, Alan Wall, Philip Brooks, and Jin Wang. Automatic Identification System (AIS): Data Reliability and Human Error Implications. *Journal of Navigation*, 60(03):373, August 2007. ISSN 1469-7785. doi: 10.1017/s0373463307004298. URL http://dx.doi.org/10.1017/S0373463307004298.

[21] Havenbedrijf Rotterdam. Jaarverslag 2014: In transitie, 2014. URL https://www.portofrotterdam.com/nl/file/1457/download?token=pvzOTCLZ.

[22] R. Heij. Opportunities for peak shaving electricity consumption at container terminals. Master's thesis, Delft University of Technology, 2015. URL http://repository.tudelft.nl/islandora/object/uuid%3A496725f4-ff7a-4319-bcaf-9e258e6dfe87?collection=education.

[23] Sascha Hornauer and Axel Hahn. Towards Marine Collision Avoidance Based on Automatic Route Exchange. *IFAC Proceedings Volumes*, 46(33):103–107, 2013. ISSN 1474-6670. doi: 10.3182/20130918-4-jp-3022.00049. URL http://www.sciencedirect.com/science/article/pii/S1474667016461410. 9th {IFAC} Conference on Control Applications in Marine Systems.

[24] Gudrun K. Høye, Torkild Eriksen, Bente J. Meland, and Bjørn T. Narheim. Space-based AIS for global maritime traffic monitoring. *Acta Astronautica*, 62(2-3):240–245, January 2008. doi: 10.1016/j.actaastro.2007.07.001.

[25] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, October 2006. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2006.03.001. URL http://dx.doi.org/10.1016/j.ijforecast.2006.03.001.

[26] International Maritime Organization. Guidelines for the onboard operational use of shipborne Automatic Indentification Systems (AIS), January 2002. URL https://www.navcen.uscg.gov/pdf/AIS/IMO_A_917(22)_AIS_OPS_Guidelines.pdf.

[27] International Maritime Organization. About imo, 2017. URL http://www.imo.org/en/About/Pages/Default.aspx.

[28] International Telecomunication Union. Recommendation itu-r m.1371-5: Technical characteristics for an automatic identification system using time division multiple access in the vhf maritime mobile frequency band, 2014.

[29] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 103 of *Springer Texts in Statistics*. Springer New York, 2013.

[30] Pan Jiacai, Jiang Qingshan, Hu Jinxing, and Shao Zheping. An AIS data Visualization Model for Assessing Maritime Traffic Situation and its Applications. *Procedia Engineering*, 29:365–369, 2012. ISSN 1877-7058. doi: 10.1016/j.proeng.2011.12.724. URL http://dx.doi.org/10.1016/j.proeng.2011.12.724.

[31] F. Katsilieris, P. Braca, and S. Coraluppi. Detection of malicious AIS position spoofing by exploiting radar information. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1196–1203, July 2013.

[32] K. Kowalska and L. Peel. Maritime anomaly detection using gaussian process active learning. In *2012 15th International Conference on Information Fusion*, pages 1164–1171, 2012.

[33] Miroslav Kubat. *An introduction to machine learning*. Springer, 2015. doi: 978-3-319-20010-1.

[34] Ove Daae Lampe, Johannes Kehrer, and Helwig Hauser. Visual analysis of multivariate movement data using interactive difference views. In *VMV*, volume 10, pages 315–322, 2010.

[35] R O Lane, D A Nevell, S D Hayward, and T W Beaney. Maritime anomaly detection and threat assessment. In *2010 13th International Conference on Information Fusion*, pages 1–8. Institute of Electrical and Electronics Engineers (IEEE), July 2010. doi: 10.1109/icif.2010.5711998.

[36] Therese Langer and Shruti Vaidyanathan. Smart Freight: Applications of Information and Communications Technologies to Freight System Efficiency. *Washington, DC: American Council for an Energy-Efficient Economy*, 2014.

[37] R Laxhammar. *Anomaly Detection in Trajectory Data for Surveillance Applications*. PhD thesis, Örebro University, 2011.

[38] R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In *2009 12th International Conference on Information Fusion*, pages 756–763, 2009.

[39] Hau L. Lee, V. Padmanabhan, and Seungjin Whang. Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558, 1997. doi: 10.1287/mnsc.43.4.546. URL http://dx.doi.org/10.1287/mnsc.43.4.546.

[40] Po-Ruey Lei, Jiunn Su, Wen-Chih Peng, Wei-Yu Han, and Chien-Ping Chang. A framework of moving behavior modeling in the maritime surveillance. *Journal of Chung Cheng Institute of Technology*, 40(2):33–42, 2011.

[41] Changqing Liu and Xiaoqian Chen. Vessel Track Recovery With Incomplete AIS Data Using Tensor CANDECOM/PARAFAC Decomposition. *Journal of Navigation*, 67(01): 83–99, July 2013. doi: 10.1017/s0373463313000398. URL https://www.cambridge.org/core/article/div-class-title-vessel-track-recovery-with-incomplete-ais-data-using-tensor-candecom-parafac-decomposition-div/15E9D39D6D77A39A290A92252EE4B462.

[42] U. Löptien and L. Axell. Ice and AIS: ship speed data and sea ice forecasts in the Baltic Sea. *The Cryosphere*, 8(6):2409–2418, December 2014. ISSN 1994-0424. doi: 10.5194/tc-8-2409-2014. URL http://dx.doi.org/10.5194/tc-8-2409-2014.

[43] She Xiang Ma, Jin Sun, and Yong Qiang Guan. Detection probability of airborne ais. In *Applied Mechanics and Materials*, volume 401, pages 1204–1207. Trans Tech Publ, 2013. ISBN 303785846X.

[44] I Menger. Information Exchange between Deep Sea Container Terminals and Hinterland Parties. Master's thesis, Delft University of Technology, 2016. URL http://repository.tudelft.nl/islandora/object/uuid%3Adf65f8c2-3c27-43ce-b9a3-768d964eef51?collection=education.

[45] Tom Minka. Regression trees, October 2001. URL http://alumni.media.mit.edu/~tpminka/courses/36-350.2001/lectures/day19/.

[46] Nadereh Moini, Maria Boile, Sotiris Theofanis, and William Laventhal. Estimating the determinant factors of container dwell times at seaports. *Maritime Economics & Logistics*, 14(2):162–177, June 2012. ISSN 1479-294X. doi: 10.1057/mel.2012.3. URL http://dx.doi.org/10.1057/mel.2012.3.

[47] M Flavia Monaco, Luigi Moccia, and Marcello Sammarra. Operations Research for the management of a transhipment container terminal: The Gioia Tauro case. *Maritime Economics & Logistics*, 11(1):7–35, March 2009. ISSN 1476-0592. doi: 10.1057/mel.2008.21. URL http://dx.doi.org/10.1057/mel.2008.21.

[48] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, March 2001. ISSN 0360-0300. doi: 10.1145/375360.375365. URL http://doi.acm.org/10.1145/375360.375365.

[49] Navigation Center. Class a ais position report (messages 1, 2, and 3), september 2016. URL https://www.navcen.uscg.gov/?pageName=AISMessagesA.

[50] Navigation Center. Ais class a ship static and voyage related data (message 5), december 2016. URL https://www.navcen.uscg.gov/?pageName=AISMessagesAStatic.

[51] David Nevell. Anomaly detection in white shipping. *Mathematics in Defence*, 2009.

[52] Andy Norris. AIS Implementation–Success or Failure? *Journal of Navigation*, 60(01):1–10, 2007.

[53] Commission of the European Communities. Common position adopted by the council with a view to the adoption of a directive of the european parliament and of the council amending directive 2002/59/ec establishing a community vessel traffic monitoring and information system, document com 2008 310 final–2005/0239 cod, 2008. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2008:0310:FIN:EN:pdf.

[54] Orbcomm. Global satelite AIS Services, 2015. URL https://www.orbcomm.com/PDF/datasheet/Satellite-AIS.pdf.

[55] Rita Osadchy. Nonparametric density estimation nearest neighbors , KNN, 2013. URL http://www.cs.haifa.ac.il/~rita/ml_course/lectures/KNN.pdf.

[56] G Pallotta, M Vespe, and K Bryan. Traffic route extraction and anomaly detection (TREAD): Vessel pattern knowledge discovery and exploitation for maritime situational awareness. *NATO Formal Report CMRE-FR-2013-001, NATO Unclassified*, 2013.

[57] Giuliana Pallotta, Michele Vespe, and Karna Bryan. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy*, 15(6):2218–2245, June 2013. ISSN 1099-4300. doi: 10.3390/e15062218. URL http://dx.doi.org/10.3390/e15062218.

[58] I Parolas. ETA prediction for containerships at the Port of Rotterdam using Machine Learning Techniques. Master's thesis, Delft University of Technology, 2016. URL http://repository.tudelft.nl/islandora/object/uuid%3A9e95d11f-35ba-4a12-8b34-d137c0a4261d?collection=education.

[59] M. Pisano. *A decision support system for planning operations at a transshipment terminal container*. PhD thesis, University of Cagliari, 2008.

[60] Martin Redoutey, Eric Scotti, Christian Jensen, Cyril Ray, and Christophe Claramunt. *Efficient Vessel Tracking with Accuracy Guarantees*, pages 140–151. Springer Nature, Berlin, Heidelberg, 2008. ISBN 978-3-540-89903-7. doi: 10.1007/978-3-540-89903-7_13. URL http://dx.doi.org/10.1007/978-3-540-89903-7_13.

[61] Bradley J. Rhodes, Neil A. Bomberger, and Majid Zandipour. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness. In *2007 10th International Conference on Information Fusion*, pages 1–8. Institute of Electrical and Electronics Engineers (IEEE), July 2007. doi: 10.1109/icif.2007.4408127.

[62] B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. In *2008 11th International Conference on Information Fusion*, pages 1–7, June 2008.

[63] Jean-Paul Rodrigue, Michael Browne, R Knowles, Jon Shaw, and Iain Docherty. International maritime freight transport and logistics. *Transport Geographies: An Introduction*, pages 156–178, 2002.

[64] Richard J Roiger. *Data mining: A tutorial-based primer*. Addison Wesley, 2002.

[65] Premalatha Sampath. Trajectory analysis using Automatic Identification System (AIS) in New Zealand waters. Master's thesis, Auckland University of Technology, 2012.

[66] Krispijn A. Scholte. Detecting Suspicious Behavior in Marine Traffic using the Automatic Identification System. Master's thesis, Delft University of Technology, 2013.

[67] Giovanni Seni and John F. Elder. Ensemble methods in data mining: Improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–126, 2010. doi: 10.2200/S00240ED1V01Y200912DMK002. URL https://doi.org/10.2200/S00240ED1V01Y200912DMK002.

[68] Yossi Sheffi. Supply chain management under the threat of international terrorism. *The International Journal of Logistics Management*, 12(2):1–11, 2001. doi: 10.1108/09574090110806262. URL http://dx.doi.org/10.1108/09574090110806262.

[69] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. ISSN 1573-1375. doi: 10.1023/B:STCO.0000035301.49549.88. URL http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

[70] Gunnar Stefansson and Johan Woxenius. The Concept of Smart Freight Transport Systems–the road haulier's perspective. In *proceedings of the 19th NOFOMA Conference, Reykjavik*, pages 7–8, 2007.

[71] Martin Stopford. *Maritime economics 3e*. Routledge, 2009.

[72] Alejandro Talavera, Ricardo Aguasca, Blas Galván, and Andrés Cacereño. Application of Dempster–Shafer theory for the quantification and propagation of the uncertainty caused by the use of AIS data. *Reliability Engineering & System Safety*, 111:95–105, March 2013. ISSN 0951-8320. doi: 10.1016/j.ress.2012.10.007. URL http://www.sciencedirect.com/science/article/pii/S0951832012002074.

[73] The Nautical Institute. AIS Initialisation, 2005. URL http://www.nautinst.org/en/forums/mars/mars-2005.cfm/AISInitialisation.

[74] Ming-Cheng Tsou. Discovering Knowledge from AIS Database for Application in VTS. *Journal of Navigation*, 63(03):449–469, May 2010. ISSN 1469-7785. doi: 10.1017/s0373463310000135. URL http://dx.doi.org/10.1017/S0373463310000135.

[75] Min Han Tun, Graeme S Chambers, Tele Tan, and Thanh Ly. Maritime port intelligence using AIS data. *Recent advances in security technology*, page 33, 2007.

[76] United Nations Conference on Trade and Development. Review of Maritime Transport 2016. Technical report, United Nations, 2016.

[77] Mark van der Loo. Package 'stringdist', 2016. URL https://cran.r-project.org/web/packages/stringdist/stringdist.pdf.

[78] Mark PJ Van der Loo. The stringdist package for approximate string matching. *The R*, 2014.

[79] T van der Schelde. Using Game Structuring Methods to Assess the Reliability of the Departure Order of Sea-Going Container Ships. Master's thesis, Delft University of Technology, 2015. URL http://repository.tudelft.nl/islandora/object/uuid:518d51ac-647e-44a0-b834-e30432f4ff8d?collection=education.

[80] M.P.A. van Oosterhout, M. Zielinski, and Y.-H. Tan. Virtuele haven deliverable t2.d1a. Technical report, 2000.

[81] Albert Veenstra, Rob Zuidwijk, and Eelco van Asperen. The extended gate concept for container terminals: Expanding the notion of dry ports. *Maritime Economics & Logistics*, 14(1):14–32, March 2012. ISSN 1476-0592. doi: 10.1057/mel.2011.15. URL http://dx.doi.org/10.1057/mel.2011.15.

[82] M Vespe, G Pallotta, I Visentini, K Bryan, and P Braca. Maritime anomaly detection based on historical trajectory mining. In *Proceedings of the NATO Port and Regional Maritime Security Symposium*, pages 1–11, 2012.

[83] M. Vespe, I. Visentini, K. Bryan, and P. Braca. Unsupervised learning of maritime traffic patterns for anomaly detection. In *9th IET Data Fusion Target Tracking Conference (DF TT 2012): Algorithms Applications*, pages 1–5. Institution of Engineering and Technology (IET), May 2012. doi: 10.1049/cp.2012.0414.

[84] Michele Vespe, Massimo Sciotti, Fabrizio Burro, Giulia Battistello, and Stefano Sorge. Maritime multi-sensor data association based on geographic and navigational knowledge. In *2008 IEEE Radar Conference*, pages 1–6. Institute of Electrical and Electronics Engineers (IEEE), May 2008. doi: 10.1109/radar.2008.4720782.

[85] Iris F.A. Vis and René de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1–16, May 2003. ISSN 0377-2217. doi: 10.1016/s0377-2217(02)00293-x. URL http://www.sciencedirect.com/science/article/pii/S037722170200293X.

[86] Stefan Voß, Robert Stahlbock, and Dirk Steenken. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1):3–49, January 2004. ISSN 1436-6304. doi: 10.1007/s00291-003-0157-z. URL http://dx.doi.org/10.1007/s00291-003-0157-z.

[87] Yang Wang, Jinfen Zhang, Xianqiao Chen, Xiumin Chu, and Xinping Yan. A spatial–temporal forensic analysis for inland–water ship collisions using AIS data. *Safety Science*, 57:187–202, August 2013. ISSN 0925-7535. doi: 10.1016/j.ssci.2013.02.006. URL http://www.sciencedirect.com/science/article/pii/S0925753513000465.

[88] Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014.

[89] Wayan Mahardhika Wijaya and Yasuhiro Nakamura. Predicting Ship Behavior Navigating through Heavily Trafficked Fairways by Analyzing AIS Data on Apache HBase. *2013 First International Symposium on Computing and Networking*, December 2013. doi: 10.1109/candar.2013.39. URL http://dx.doi.org/10.1109/CANDAR.2013.39.

[90] Rob A. Zuidwijk. Are we connected?, November 2015.

[91] Rob A. Zuidwijk and Albert W. Veenstra. The Value of Information in Container Transport: Leveraging the Triple Bottom Line. *ERIM report series research in management Erasmus Research Institute of Management*, (ERS-2010-039-LIS), October 2010.

[92] Rob A. Zuidwijk and Albert W. Veenstra. The Value of Information in Container Transport. *Transportation Science*, 49(3):675–685, August 2015. doi: 10.1287/trsc.2014.0518. URL http://dx.doi.org/10.1287/trsc.2014.0518.

A

Overview of AIS related research

Table A.1: Overview of AIS related researches

Authors	Year	Title
S.J. Chang	2004	Development and analysis of AIS applications as an efficient tool for vessel traffic service[8].
A. Graveson	2004	AIS - An Inexact Science [16]
N.J. Bailey	2005	Training, technology and AIS: looking beyond the box[4]
N. Bomberger et al.	2006	Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness[6]
A. Harati-Mokhtari et al.	2007	Automatic Identification System (AIS): Data Reliability and Human Error Implications[20]
A. Norris	2007	AIS Implementation–Success or Failure?[52]
B.J. Rhodes et al.	2007	Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness[61]
M.H. Tun et al.	2007	Maritime port intelligence using AIS data[75]
M. Baldauf et al.	2008	Aspects of technical reliability of navigation systems and human element in case of collision avoidance[5]
G.K. Høye et al.	2008	Space-based AIS for global maritime traffic monitoring[24]
M. Redoutey et al.	2008	Efficient Vessel Tracking with Accuracy Guarantees[60]
B. Ristic et al.	2008	Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction[62]
M. Vespe et al.	2008	Maritime multi-sensor data association based on geographic and navigational knowledge[84]
K.G. Aarsæther & T. Moan	2009	Estimating Navigation Patterns from AIS[1]
R. Laxhammer et al.	2009	Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator [38]
T. de Boer	2010	An analysis of vessel behaviour based on AIS data[9]
U. Demšar & K. Virrantaus	2010	Space–time density of trajectories: exploring spatio-temporal patterns in movement data[11]
M. Guerriero et al.	2010	Analysis of AIS Intermittency and Vessel Characterization using a Hidden Markov Model [17]
O.D. Lampe et al.	2010	Visual analysis of multivariate movement data using interactive difference views[34]
R.O. Lane et al.	2010	Maritime anomaly detection and threat assessment[35]
M. Tsou	2010	Discovering Knowledge from AIS Database for Application in VTS[74]

Continued on next page

Table A.1 – continued from previous page

Authors	Year	Title
R. Laxhammer	2011	Anomaly Detection in Trajectory Data for Surveillance Applications[37]
P. Lei et al.	2011	A framework of moving behavior modeling in the maritime surveillance[40]
P. Jiacai et al.	2012	An AIS data Visualization Model for Assessing Maritime Traffic Situation and its Applications[30]
K. Kowalska & L. Peel	2012	Maritime anomaly detection using Gaussian Process active learning[32]
P. Sampath	2012	Trajectory analysis using Automatic Identification System (AIS) in New Zealand waters[65]
M. Vespe et al.	2012	Maritime anomaly detection based on historical trajectory mining[82]
M. Vespe et al.	2012	Unsupervised learning of maritime traffic patterns for anomaly detection[83]
S. Hornauer & A. Hahn	2013	Towards Marine Collision Avoidance Based on Automatic Route Exchange[23]
F. Katsilieris et al.	2013	Detection of malicious AIS position spoofing by exploiting radar information[31]
C. Liu & X. Chen	2013	Vessel Track Recovery With Incomplete AIS Data Using Tensor CANDECOM/PARAFAC Decomposition[41]
S.X. Ma et al.	2013	Detection probability of airborne AIS[43]
G. Pallotta et al.	2013	Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction[57]
K.A. Scholte	2013	Detecting Suspicious Behavior in Marine Traffic using the Automatic Identification System[66]
A. Talavera et al.	2013	Application of Dempster-Shafer theory for the quantification and propagation of the uncertainty caused by the use of AIS data[72]
Y. Wang et al.	2013	A spatial–temporal forensic analysis for inland–water ship collisions using AIS data [87]
W.M. Wijaya & Y. Nakamura	2013	Predicting Ship Behavior Navigating through Heavily Trafficked Fairways by Analyzing AIS Data on Apache HBase[89]
U. Löptien & L. Axell	2014	Ice and AIS: ship speed data and sea ice forecasts in the Baltic Sea[42]
A. Dobrkovic et al.	2015	Towards an Approach for Long Term AIS-Based Prediction of Vessel Arrival Times[12]
I. de Vreede	2016	Managing Historic Automatic Identification System data by using a proper Database Management System structure[10]
I. Parolas	2016	ETA prediction for containerships at the Port of Rotterdam using Machine Learning Techniques[58]

B

Overview of faulty MMSI numbers

Table B.1: Overview of vessels with multiple MMSI numbers

MMSI	IMO	Name
214182063,214182603,373381000,636015669	8324593	RIVER PRIDE
310665000,636016907,636016936	8902565	KALLIOPI R.C.
477382000,636017037	9108166	BOX HONGKONG
304418000,667001412	9121883	JAOHAR RIMA
244870287,304939000	9122241	A2B FUTURE
242198100,636091020	9141792	CIELO DI RABAT
236362000,258316000	9144689	NOR FEEDER
255805759,304116000	9164550	MARIA P
232613000,636016979	9189354	MV DIMITRIS Y
230942790,304140000	9197478	WIEBKE
477830600,563460000	9215878	MAERSK NEWCASTLE
477830500,563458000	9215907	MARIANNE
224505000,255805763	9216858	OPDR CADIZ
211425360,255805677	9221827	BUXCOAST
211378777,211378810	9222273	CONTI PARIS
218082000,636017022	9224049	MSC SHIRLEY
538090289,636092637	9225433	AS VEGA
305739000,636092635	9226372	MAIKE D
477036600,636091155	9231248	E.R. INDIA
209627000,304080000	9234989	AURORA
226320000,253201000	9256365	DURANDE
357249000,563791000	9261712	MOL ENCORE
352987000,563557000	9261724	AL ENDEAVOR
355109000,563788000	9261736	AL ENDURANCE
636016851,636090705	9275036	BOMAR JULIANA
256639000,256639016	9277400	OELAND
636091113,636091123	9286774	RIO TAKU
235775000,256677000	9292943	X-PRESS MONTECERVINO
636091116,636091328	9295373	MSC SHANGHAI
255805779,636091048	9295945	ALLEGORIA
255805683,636090822	9301445	E.R. CALAIS
212675000,244850970	9302243	SVEN D
212644000,244850968	9302255	SPIRIT

Continued on next page

Table B.1 – continued from previous page

MMSI	IMO	Name
240475000,256940000	9305570	COSCO GUANGZHOU
240499000,256938000	9305582	COSCO NINGBO
240513000,256930000	9305594	COSCO YANTIAN
240512000,256937000,256937009	9308508	COSCO BEIJING
240511000,256932000	9308510	EBAHELLAS
354314974,354315000	9309461	MSC TOMOKO
477690700,477690744	9314234	CSCL ZEEBRUGGE
229384000,255805617	9320398	MSC LAUSANNE
635961393,635961485,636016199	9332872	ZIM HAMBURG
229821000,229821007	9337597	GALANI
636017112,636091342	9339583	POMERENIA SKY
538090304,538090400	9344722	VALENTINA
236111837,305195000	9345972	MSC ATLANTA
210248000,210248096,210248160	9349227	ASTRORUNNER
244740921,256381000	9360582	MULTRATUG 4
304071000,306849000	9369083	MARMACTAN
218760000,636016975	9395551	SEABOARD PATRIOT
229309000,636016973	9408774	BOMAR RESOLUTE
305258993,305259000	9433444	MAX PRODUCER
255805755,305817000	9436305	JOHANNA SCHEPERS
256604000,256687000	9436379	CMA CGM SAMSON
255805796,636091922	9437050	BARBARA
305360000,636092616	9437191	A S FATIMA
255805799,636092166	9447861	MSC FILOMENA
566318977,566319000	9461879	APL GWANGYANG
477397800,477397812	9472177	COSCO HARMONY
209467000,211242470	9483671	NORDIC STANI
245930984,245931000	9507051	BRENT
256576000,447171000	9525924	AL QIBLA
218790986,218791000	9612997	ANTWERPEN EXPRESS
241239000,256858000	9618305	MSC ATHENS
241240000,256871000	9618317	MSC ATHOS
253056000,255805698	9622203	CAP SAN NICOLAS
253346000,255805696	9622215	CAP SAN MARCO
253126000,255805699	9622227	CAP SAN LORENZO
477300000,477776100,477776122	9630365	HANJIN BUDDHA
563234000,563234002	9632026	MOL QUASAR
564387000,564573000	9633941	CAP SAN MALEAS
234567891,256213000	9674567	CMA CGM THAMES
477711700,477967800	9695145	CSCL ATLANTIC OCEAN
447705965,477967700	9695169	CSCL ARCTIC OCEAN
214000000,235108381	9702132	CMA CGM KERGUELEN
477737600,477737700	9704623	YM WELLNESS
234567891,235111246	9706889	CMACGM VASCO DE GAMA

C

R code

```r
library(readr)
library(xtable)
library(dplyr)
library(ggplot2)
library(maps)
library(ggmap)
library(mapproj)
options(stringsAsFactors = FALSE)


#################################Data exploration#####################################

#Read in data and check structure and data
ERP_AIS <- read_csv("ERP_AIS.csv",
                    col_types = cols(
                      eta = col_datetime(format = "%d/%m/%Y %H:%M"),
                      timestamp = col_datetime(format = "%d/%m/%Y %H:%M")
                    ))
View(ERP_AIS)
glimpse(ERP_AIS)
head(ERP_AIS, n = 10)
summary(ERP_AIS)

#Plot datapoints to world map
mp <- NULL
mapWorld <- borders("world", colour="gray50", fill="gray50") # create a layer of borders
mp <- ggplot() + mapWorld
mp <- mp + geom_point(data = ERP_AIS, aes(x = longitude, y = latitude), color = "blue", shape = ".")
mp

#In-depth look at variables
unique(ERP_AIS$type)
length(unique(ERP_AIS$mmsi))
length(unique(ERP_AIS$imo))
length(unique(ERP_AIS$name))
```

```
combinations <- unique(ERP_AIS[c("mmsi", "imo", "name")])
View(combinations)

################################Data cleaning####################################

#This function check if ships have passed the area at the port of Rotterdam.
# After this check I manually checked if ships were missed. All those ships are
# added to a dataframe and filtered by type so only container vessels remain.

clean_ships <- function(ERP_AIS) {
  #Generate Vector of IMO that visit Rotterdam and store these in a new dataset
  ERP_AIS_imo <-
    unique(ERP_AIS$imo[ERP_AIS$latitude > 51.95 & ERP_AIS$latitude < 52.0 & ERP_AIS$longitude > 4.05 & ERP_AIS$longitude < 4.1 & ERP_AIS$course < 180])
  ERP_AIS_clean <- ERP_AIS %>%
    filter(imo %in% ERP_AIS_imo)

  #IMO of ships that should be in dataset
  add <- 8209731
  add <- c(add, 9156199)
  add <- c(add, 9168843)
  add <- c(add, 9193240)
  add <- c(add, 9261451)
  add <- c(add, 9277395)
  add <- c(add, 9307243)
  add <- c(add, 9312810)
  add <- c(add, 9313199)
  add <- c(add, 9314246)
  add <- c(add, 9339583)
  add <- c(add, 9349186)
  add <- c(add, 9395161)
  add <- c(add, 9429273)
  add <- c(add, 9461465)
  add <- c(add, 9462706)
  add <- c(add, 9604081)
  add <- c(add, 9604160)
  add <- c(add, 9629093)
  add <- c(add, 9629902)
  add <- c(add, 9645920)
  add <- c(add, 9674567)
  add <- c(add, 9685358)
  add <- c(add, 9704611)
  add <- c(add, 9215878)
  add <- c(add, 9305582)
  add <- c(add, 9629031)
  add <- c(add, 9713349)
  add <- c(add, 9301445)
  add <- c(add, 9702156)
  add <- c(add, 9200677)
```

```
  #Add ships that did visit rotterdam to new dataset
  ERP_AIS_clean <- ERP_AIS %>%
    filter(ERP_AIS$imo %in% add) %>%
    bind_rows(ERP_AIS_clean)

  #clean by vessel type
  ERP_AIS_clean <- filter(ERP_AIS_clean, type == "container ship" |
    ↪ type == "container ship (fully cellular)")

  #Remove vessels with a ETA before 01-01-1970 00:00
  ERP_AIS_clean <- filter(ERP_AIS_clean, eta > 0)

  return(ERP_AIS_clean)
}
ERP_AIS_clean <- clean_ships(ERP_AIS)

#plot datapoint to worldmap
mp <- ggplot() + mapWorld
mp + geom_point(data = ERP_AIS_clean, aes(x = longitude, y =
  ↪ latitude), color = "blue", shape = ".")



################################Data
  ↪ manipulation#######################################

#Reads in a database from a file I found on the internet with
  ↪ location info of ports
read_WPI <- function() {
  WPI <-read_delim("~/Dropbox/Master
    ↪ Thesis/Data4TU/WPI.csv",";",escape_double = FALSE,trim_ws =
    ↪ TRUE)
  WPI$Latitude <- NA
  WPI$Longitude <- NA
  for (i in 1:nrow(WPI)) {
    if (WPI$Latitude_hemispehere[i] == "S")
      WPI$Latitude_degress[i] <- WPI$Latitude_degress[i] * -1
    if (WPI$Longitude_hemisphere[i] == "W")
      WPI$Longitude_degrees[i] <- WPI$Longitude_degrees[i] * -1
    WPI$Latitude[i] <- as.numeric(paste(WPI$Latitude_degress[i],
      ↪ WPI$Latitude_minutes[i], sep ="."))
    WPI$Longitude[i] <- as.numeric(paste(WPI$Longitude_degrees[i],
      ↪ WPI$Longitude_minutes[i], sep ="."))
  }
  return(WPI)
}
WPI <- read_WPI()

# Some ships do not change status to 5(moored) when they are being
  ↪ (off)loaded. This function
# changes every ships that is travelling less then 500m per hour near
  ↪ a port to status 5. and vice
# versa
change_status <- function(ERP_AIS_clean) {
```

```
  pb <-txtProgressBar(min = 0, max = nrow(ERP_AIS_clean),style = 3)

  for (i in 1:nrow(ERP_AIS_clean)) {
    setTxtProgressBar(pb, i)

    if (ERP_AIS_clean$status[i] == 0 && ERP_AIS_clean$speed[i] <=
      ↪ 0.5) {
      for (j in 1:nrow(WPI)) {
        if (abs(ERP_AIS_clean$longitude[i] - WPI$Longitude[j]) < 1 &&
          ↪ abs(ERP_AIS_clean$latitude[i] - WPI$Latitude[j]) < 1) {
          ERP_AIS_clean$status[i] <- 5
          break
        }
      }
    }

    else if (ERP_AIS_clean$speed[i] > 0.5 && ERP_AIS_clean$status[i]
      ↪ == 5)
      ERP_AIS_clean$status[i] <- 0
  }

  return(ERP_AIS_clean$status)
}
ERP_AIS_clean$status <- change_status(ERP_AIS_clean)

# This function arranges the dataset by imo and timestamp. Then it
  ↪ loops over the database.
# If a ship gets status 5 in the Port of Rotterdam we will set that
  ↪ observation as the actual time
# of arrival. Give the stop a route ID and calulate the estimated
  ↪ traveltime by subtracting the time
# of the observation from the ETA. Then with a while loop we also set
  ↪ in all the preceding
# observations of the journey the ATA, route ID and estimated
  ↪ traveltime.
calculate_ATA_route_est_travel <- function(ERP_AIS_clean) {
  ERP_AIS_clean <- arrange(ERP_AIS_clean, imo, timestamp)
  ERP_AIS_clean$ATA <- NA
  ERP_AIS_clean$route_id <- NA
  ERP_AIS_clean$est_traveltime <- NA
  route <- 1
  max <- nrow(ERP_AIS_clean)

  pb <- txtProgressBar(min = 0,max = max,style = 3,title = "Assigning
    ↪ ATA & Route ID, calculating estimated traveltime")

  #Set ATA and routes
  i <- 1
  while (i <= max) {
    setTxtProgressBar(pb, i)
    #Set ATA & route_id
    if ((i - 1 > 0 && ERP_AIS_clean$imo[i] != ERP_AIS_clean$imo[i -
      ↪ 1])) {
      route <- 1
    }
```

```
175
      if (ERP_AIS_clean$latitude[i] > 51.85 &&
        ↪ ERP_AIS_clean$latitude[i] < 52.0 &&
        ↪ ERP_AIS_clean$longitude[i] > 3.95 &&
        ↪ ERP_AIS_clean$longitude[i] < 4.6 && ERP_AIS_clean$status[i]
        ↪ == 5) {
        ERP_AIS_clean$ATA[i] <- ERP_AIS_clean$timestamp[i]
        ERP_AIS_clean$route_id[i] <- route
        ERP_AIS_clean$est_traveltime[i] <-
          ↪ difftime(ERP_AIS_clean$eta[i], ERP_AIS_clean$timestamp[i],
          ↪ tz = "GMT", units = "hours")
180
        k <- 1
        while (i - k > 0 && ERP_AIS_clean$imo[i] == ERP_AIS_clean$imo[i
          ↪ - k] && is.na(ERP_AIS_clean$ATA[i - k])) {
          index <- i - k
          ERP_AIS_clean$ATA[index] <- ERP_AIS_clean$ATA[i]
185       ERP_AIS_clean$route_id[index] <- route
          ERP_AIS_clean$est_traveltime[index] <-
            ↪ difftime(ERP_AIS_clean$eta[i],
            ↪ ERP_AIS_clean$timestamp[index], tz = "GMT", units =
            ↪ "hours")
          k <- k + 1
        }


190
        l <- 1
        while (i + l < max && ERP_AIS_clean$status[i + l] == 5) {
          ERP_AIS_clean$ATA[i+l] <- 1
          l <- l + 1
195       }
        i <- i + l
        route <- route + 1
      }

200   else
        i <- i + 1
    }
    return(ERP_AIS_clean)
  }
205 ERP_AIS_clean <- calculate_ATA_route_est_travel(ERP_AIS_clean)

  #change class of ATA to POSIXct
  ERP_AIS_clean$ATA <- as.POSIXct(ERP_AIS_clean$ATA, tz = "GMT", origin
    ↪ = "1970-01-01 00:00:00")

210 ############Backup 1#################################
  ERP_AIS_Backup <- ERP_AIS_clean

  #Remove observations if ships do not return to Rotterdam and
    ↪ observations while moored in Rotterdam
  ERP_AIS_relevant <- ERP_AIS_clean[!is.na(ERP_AIS_clean$ATA),]
215 ERP_AIS_relevant <- ERP_AIS_relevant[ERP_AIS_relevant$ATA != 1,]
```

```r
# Some ships have the same ETA for multiple journeys. So we remove journeys that have the ETA
# of other journeys. We only keep the journey with the best ETA prediction.
remove_overlap_ETA <- function(ERP_AIS_relevant) {
  i <- 1
  pb <-txtProgressBar(min = 0, max = nrow(ERP_AIS_relevant), style = 3, title = "Remove overlapping ETA")

  remove <- data.frame(imo = numeric(0), route_id = numeric(0))
  while (i < nrow(ERP_AIS_relevant)) {
    setTxtProgressBar(pb, i)

    if (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i + 1] &&
      ERP_AIS_relevant$route_id[i] != ERP_AIS_relevant$route_id[i + 1]) {
      j <- 2
      while (ERP_AIS_relevant$route_id[i + 1] ==
        ERP_AIS_relevant$route_id[i + j] && i + j <=
        nrow(ERP_AIS_relevant))
        j <- j + 1
      if (ERP_AIS_relevant$eta[i] == ERP_AIS_relevant$eta[i + j - 1]
        && abs(ERP_AIS_relevant$est_traveltime[i]) >
        abs(ERP_AIS_relevant$est_traveltime[i + j - 1]))
        remove <- bind_rows(remove, data.frame(imo =
          ERP_AIS_relevant$imo[i], route =
          ERP_AIS_relevant$route_id[i]))

      else if (ERP_AIS_relevant$eta[i] == ERP_AIS_relevant$eta[i + j - 1])
        remove <- bind_rows(remove, data.frame(imo =
          ERP_AIS_relevant$imo[i + j - 1], route =
          ERP_AIS_relevant$route_id[i + j - 1]))

      i <- i + j - 1
    }

    else
      i <- i + 1
  }

  j <- 1
  for (i in 1:nrow(remove)) {
    setTxtProgressBar(pb, i)

    while (!(ERP_AIS_relevant$imo[j] == remove$imo[i] &&
      ERP_AIS_relevant$route_id[j] == remove$route[i]) && j <=
      nrow(ERP_AIS_relevant))
      j <- j + 1

    while (ERP_AIS_relevant$imo[j] == remove$imo[i] &&
      ERP_AIS_relevant$route_id[j] == remove$route[i] && j <=
      nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$imo[j] <- NA
      j <- j + 1
```

```
    }

  }
  ERP_AIS_relevant <- ERP_AIS_relevant[!is.na(ERP_AIS_relevant$imo),]
}
ERP_AIS_relevant <- remove_overlap_ETA(ERP_AIS_relevant)

#set treshold in hours to select ships that are maximum 2 weeks away
  ↪ from the Port of Rotterdam
traveltime_treshold <- 24 * 7 * 2

# Calculates the traveltime per observations by taking the difference
  ↪ between ATA and timestamp.
# If the traveltime is below the treshold the speed variables and
  ↪ dimensions are calculated.
calculate_traveltime_speed <- function(ERP_AIS_relevant,
  ↪ traveltime_treshold) {
  #Do other calculations/manipulation on relevant observations
  ERP_AIS_relevant$avg_speed <- NA
  ERP_AIS_relevant$obs_speed <- NA
  ERP_AIS_relevant$d_speed <- NA
  ERP_AIS_relevant$length <- NA
  ERP_AIS_relevant$width <- NA
  ERP_AIS_relevant$traveltime <- NA

  pb <-txtProgressBar(min = 0, max = nrow(ERP_AIS_relevant), style =
    ↪ 3, title = "Calculate traveltime, dimensions and speed
    ↪ variables")

  for (i in 1:nrow(ERP_AIS_relevant)) {
    setTxtProgressBar(pb, i)

    ERP_AIS_relevant$traveltime[i] <-
      ↪ difftime(ERP_AIS_relevant$ATA[i],
      ↪ ERP_AIS_relevant$timestamp[i], tz = "GMT", units = "hours")

    if (ERP_AIS_relevant$traveltime[i] <= traveltime_treshold) {
      #Calculate dimensions
      if (ERP_AIS_relevant$bow[i] > 0) {
        ERP_AIS_relevant$length[i] <- ERP_AIS_relevant$bow[i] +
          ↪ ERP_AIS_relevant$stern[i]
        ERP_AIS_relevant$width[i] <- ERP_AIS_relevant$port[i] +
          ↪ ERP_AIS_relevant$starboard[i]
      }

      #Set speed variables
      avg = ERP_AIS_relevant$speed[i]
      j <- 1
      for (j in 1:12) {
        #Speed diff over 3 hours
        if (j == 3 && i - j > 0 && ERP_AIS_relevant$imo[i] ==
          ↪ ERP_AIS_relevant$imo[i - j] &&
          ↪ ERP_AIS_relevant$route_id[i] ==
          ↪ ERP_AIS_relevant$route_id[i - j]) {
          ERP_AIS_relevant$d_speed[i] <- ERP_AIS_relevant$speed[i] -
            ↪ ERP_AIS_relevant$speed[i - j]
```

```
            }
            else if (j < 3) {
              ERP_AIS_relevant$d_speed[i] <- 0
            }

            #Avg speed
            if (i - j > 0 && ERP_AIS_relevant$imo[i] ==
              ↪ ERP_AIS_relevant$imo[i - j] &&
              ↪ ERP_AIS_relevant$route_id[i] ==
              ↪ ERP_AIS_relevant$route_id[i - j]) {
              avg <- avg + ERP_AIS_relevant$speed[i - j]
              if (j == 12) {
                ERP_AIS_relevant$avg_speed[i] <- avg / j
                ERP_AIS_relevant$obs_speed[i] <- j
              }
            }
            else{
              ERP_AIS_relevant$avg_speed[i] <- avg / j
              ERP_AIS_relevant$obs_speed[i] <- j
              break
            }
          }
        }
      }
      return(ERP_AIS_relevant)
    }
    ERP_AIS_relevant <- calculate_traveltime_speed(ERP_AIS_relevant,
      ↪ traveltime_treshold)

    ERP_AIS_Backup2 <- ERP_AIS_relevant
    ##########Made backup2##########

    #Remove observations if ships are more than the treshold away from
      ↪ the Port of Rotterdam
    ERP_AIS_relevant <- filter(ERP_AIS_relevant, traveltime <=
      ↪ traveltime_treshold)

    #Remove journeys where the elapsed time is covered by less then 90%
      ↪ by the observations,
    # very short routes, e.g. journeys in the port of rotterdam, and
      ↪ routes that have a very
    # large or negative ETA at the start of the journey. The communicated
      ↪ ETA in these cases
    # is probably from another journey.
    remove_incomplete_short_long_est <- function(ERP_AIS_relevant) {
      #Check if routes are complete
      times <- ERP_AIS_relevant %>%
        group_by(imo, route_id) %>%
        arrange(timestamp) %>%
        summarize(occurences = n(), timespan = difftime(last(timestamp),
          ↪ first(timestamp), tz = "GMT", units = "hours"),
          ↪ percentage_covered = occurences / timespan * 100, difference
          ↪ = occurences - timespan, max_estimate = max(est_traveltime))

      remove <- data.frame(imo = numeric(0), route = numeric(0))
```

```
    pb <- txtProgressBar(min = 0, max = nrow(times), style = 3, title =
      ↪ "Searching journeys to be removed")

    for (i in 1:nrow(times)) {
      setTxtProgressBar(pb, i)
      if (times$percentage_covered[i] < 90) {
        remove <- bind_rows(remove, data.frame(imo = times$imo[i],
          ↪ route = times$route_id[i]))
      }

      else if (times$occurences[i] < 5)
        remove <- bind_rows(remove, data.frame(imo = times$imo[i],
          ↪ route = times$route_id[i]))

      else if (times$max_estimate[i] > 1000)
        remove <- bind_rows(remove, data.frame(imo = times$imo[i],
          ↪ route = times$route_id[i]))

      else if (times$max_estimate[i] < 0)
        remove <- bind_rows(remove, data.frame(imo = times$imo[i],
          ↪ route = times$route_id[i]))

    }

    if (nrow(remove) == 0)
      return(ERP_AIS_relevant)

    pb <- txtProgressBar(min = 0, max = nrow(remove), style = 3, title
      ↪ = "Removing incomplete, short or journeys with wrong ETA")

    j <- 1
    for (i in 1:nrow(remove)) {
      setTxtProgressBar(pb, i)

      while (!(ERP_AIS_relevant$imo[j] == remove$imo[i] &&
        ↪ ERP_AIS_relevant$route_id[j] == remove$route[i]) && j <=
        ↪ nrow(ERP_AIS_relevant))
        j <- j + 1

      while (ERP_AIS_relevant$imo[j] == remove$imo[i] &&
        ↪ ERP_AIS_relevant$route_id[j] == remove$route[i] && j <=
        ↪ nrow(ERP_AIS_relevant)) {
        ERP_AIS_relevant$imo[j] <- NA
        j <- j + 1
      }

    }
    ERP_AIS_relevant <- ERP_AIS_relevant[!is.na(ERP_AIS_relevant$imo),]

    return(ERP_AIS_relevant)

  }
  ERP_AIS_relevant <-
    ↪ remove_incomplete_short_long_est(ERP_AIS_relevant)
```

```
# Remove journeys where a gap of at least 5 hours exist in the observations.
remove_gaps <- function(ERP_AIS_relevant) {
  remove <- data.frame(imo = numeric(0), route_id = numeric(0))
  pb <- txtProgressBar(min = 0, max = nrow(ERP_AIS_relevant), style = 3, title = "Searching journeys with large gaps to be removed")

  i <- 1
  while (i < nrow(ERP_AIS_relevant)) {
    setTxtProgressBar(pb, i)

    if (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i + 1] && ERP_AIS_relevant$route_id[i] == ERP_AIS_relevant$route_id[i + 1] && difftime(ERP_AIS_relevant$timestamp[i + 1], ERP_AIS_relevant$timestamp[i], tz = "GMT", units = "hours") > 5) {
      remove <-bind_rows(remove, data.frame(imo = ERP_AIS_relevant$imo[i], route = ERP_AIS_relevant$route_id[i]) )
      skip <- ERP_AIS_relevant$route_id[i]
      while (skip == ERP_AIS_relevant$route_id[i] && i < nrow(ERP_AIS_relevant))
        i <- i + 1
    }

    else
      i <- i + 1
  }

  if (nrow(remove) == 0)
    return(ERP_AIS_relevant)

  pb <- txtProgressBar( min = 0, max = nrow(remove), style = 3, title = "Removing journeys with gaps")
  j <- 1
  for (i in 1:nrow(remove)) {
    setTxtProgressBar(pb, i)

    while (!(ERP_AIS_relevant$imo[j] == remove$imo[i] && ERP_AIS_relevant$route_id[j] == remove$route[i] ) && j <= nrow(ERP_AIS_relevant))
      j <- j + 1

    while (ERP_AIS_relevant$imo[j] == remove$imo[i] && ERP_AIS_relevant$route_id[j] == remove$route[i] && j <= nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$imo[j] <- NA
      j <- j + 1
    }

  }
  ERP_AIS_relevant <- ERP_AIS_relevant[!is.na(ERP_AIS_relevant$imo),]
  return(ERP_AIS_relevant)
}
ERP_AIS_relevant <- remove_gaps(ERP_AIS_relevant)
```

```
# Create a database of portnames and portcodes by scraping online websites
read_ports <- function() {
  library(rvest)

  ports1 <- data.frame(1)
  while (nrow(ports1) != 27828) {
    pb <- txtProgressBar(min = 0, max = 557, style = 3, title = "Read ports from nslworld")

    for (i in 1:557) {
      setTxtProgressBar(pb, i)
      url <-paste("http://www.nslworld.net/ports.php?page=", i , sep = "")
      temp <- read_html(url)

      if (i == 1) {
        ports1 <- temp %>%
          html_nodes("table") %>%
          .[[4]] %>%
          html_table(header = T)
      }

      else
        ports1 <- bind_rows(ports1, temp %>%
                            html_nodes("table") %>%
                            .[[4]] %>%
                            html_table(header = T))
    }
    if (nrow(ports1) != 27828)
      print(paste(nrow(ports1), "ports put into database. Should be 27828 ports. Start over."))
  }


  temp <-
    read_html("https://www.marinetraffic.com/nl/ais/index/ports/all/per_page:50")
  n_ports <- temp %>%
    html_nodes("strong") %>%
    .[[6]] %>%
    html_text()

  n_ports <- as.numeric(sub(",", "", n_ports))

  ports2 <- data.frame(1)
  while (nrow(ports2) != n_ports) {
    pb <- txtProgressBar( min = 0, max = 384, style = 3, title = "Read ports from marinetraffic")

    for (i in 1:384) {
      setTxtProgressBar(pb, i)
      url <- paste(
        "https://www.marinetraffic.com/nl/ais/index/ports/all/per_page:50/page:",
        i , sep = "")
```

```
      temp <- read_html(url)

      if (i == 1) {
        ports2 <- temp %>%
          html_nodes("table") %>%
          .[[1]] %>%
          html_table(header = T)
      }

      else
        ports2 <- bind_rows(ports2, temp %>%
                            html_nodes("table") %>%
                            .[[1]] %>%
                            html_table(header = T))
    }
    if (nrow(ports2) != n_ports)
      print(paste(nrow(ports2), "ports put into database. Should be
        ↪ 19182 ports. Start over."))
  }

  ports2 <- filter(ports2, Type == "Haven")
  ports2 <- filter(ports2, `UN/LOCODE` != "-")
  ports2 <- ports2[, 1:3]
  ports2$`UN/LOCODE` <- sub("\\s", "", ports2$`UN/LOCODE`)
  ports2$temp <- ports2$Land
  ports2$Land <- NULL
  names(ports2) <- names(ports1)


  ports <- bind_rows(ports1, ports2)
  ports <- ports[order(ports$PortCode),]
  ports$PortName <- toupper(ports$PortName)
  ports$PortCode <- toupper(ports$PortCode)
  temp <- NA
  i <- 2
  pb <- txtProgressBar( min = 0, max = nrow(ports), style = 3, title
    ↪ = "Removing duplicates")

  while (i <= nrow(ports)) {
    setTxtProgressBar(pb, i)
    if (ports$PortCode[i] == ports$PortCode[i - 1]) {
      if (!is.na(ports$CountryCode[i]))
        ports <- ports[-i,]
      else
        ports <- ports[-(i - 1),]
    }
    else
      i <- i + 1
  }

  j <- 1
  k <- 1
  l <- 0
  ports <- ports[order(ports$PortCode),]
  ports_list_code <- list()
```

```
    pb <- txtProgressBar( min = 0, max = 25, style = 3, title =
      ↪ "Creating list of Port Codes")

    for (i in toupper(letters)) {
      l <- l + 1
      setTxtProgressBar(pb, l)

      while (grepl(paste0("^", i), ports$PortCode[j]))
        j <- j + 1

      ports_list_code[[i]] <- ports[k:j - 1, -3]
      k <- j + 1
    }

    j <- 1
    k <- 1
    l <- 0
    ports <- ports[order(ports$PortName),]
    ports_list_name <- list()
    while (grepl("^\\?", ports$PortName[j])) {
      j <- j + 1
      k <- j
    }
    pb <- txtProgressBar( min = 0, max = 25, style = 3, title =
      ↪ "Creating list of Port Names")
    for (i in toupper(letters)) {
      l <- l + 1
      setTxtProgressBar(pb, l)

      while (grepl(paste0("^", i), ports$PortName[j]))
        j <- j + 1

      ports_list_name[[i]] <- unique(ports[k:j - 1, 1])
      k <- j + 1
    }

    return(list(ports_list_code, ports_list_name))
  }
  ports <- read_ports()
  ports_list_code <- ports[[1]]
  ports_list_name <- ports[[2]]

  # Clean the destinations, remove digits, punctuation and alter
    ↪ notations like
  # origin -> dest or dest via nok. Remove extra whitespaces and when
    ↪ the resulting
  # dest is empty set to NA.
  clean_destination <- function(temp) {
    if (grepl("RTM>NL", temp))
      temp <- sub("RTM>NL", "NLRTM", temp)

    if (grepl(".*>+", temp))
      temp <- sub(".*>+", "", temp)

    if (grepl(",.*", temp))
```

```
575       temp <- sub(",.*", "", temp)

      if (grepl("/.*", temp))
        temp <- sub("/.*", "", temp)

580   if (grepl("..... &", temp))
        temp <- sub("..... &", "", temp)

      if (grepl("//sP//S", temp))
        temp <- sub("//sP//S", "", temp)
585
      if (grepl("[[:digit:]]", temp))
        temp <- gsub("[[:digit:]]", "", temp)

      if (grepl("[[:punct:]]", temp))
590     temp <- gsub("[[:punct:]]", " ", temp)

      if (grepl("VIA\\s.*", temp))
        temp <- sub("VIA\\s.*", "", temp)

595   if (grepl("VA\\s.*", temp))
        temp <- sub("VA\\s.*", "", temp)

      if (grepl(".*\\sTO", temp))
        temp <- sub(".*\\sTO", "", temp)
600
      if (grepl("\\sBY\\S", temp))
        temp <- sub("\\sBY\\S", " ", temp)

      if (grepl("SHIFTING", temp))
605     temp <- sub("SHIFTING", " ", temp)

      if (grepl("TO ORDER", temp))
        temp <- sub("TO ORDER", " ", temp)

610   if (grepl("\\sBERTH", temp))
        temp <- sub("\\sBERTH", " ", temp)

      if (grepl("FOR\\s", temp))
        temp <- sub("FOR\\s", "", temp)
615
      if (grepl("\\sORDERS", temp))
        temp <- sub("\\sORDERS", "", temp)

      if (grepl("\\sORDER", temp))
620     temp <- sub("\\sORDER", "", temp)

      if (grepl("\\sPILOT", temp))
        temp <- sub("\\sPILOT", "", temp)

625   if (grepl("MOORED", temp))
        temp <- sub("MOORED", "", temp)

      if (grepl("ANCHORAGE", temp))
        temp <- sub("ANCHORAGE", "", temp)
```

```
630
    if (grepl("SHIFT", temp))
      temp <- sub("SHIFT", "", temp)

    if (grepl("\\s+", temp))
635   temp <- gsub("\\s+", " ", temp)

    temp <- trimws(temp, "both")

    if (grepl("^..\\s...$", temp))
640   temp <- sub("\\s", "", temp)

    if (nchar(temp) == 0)
      temp <- NA

645 return(temp)
  }

  pb <- txtProgressBar( min = 0, max = nrow(ERP_AIS_relevant), style =
    ↪ 3, title = "Cleaning destinations")

650 # Call clean_destination for every first appearance of a destination.
    ↪ Then for every following
  # appearance of the same destination in the same journey assign same
    ↪ value. When the destination is
  # already changed between the last sailing observation and the next
    ↪ moored observations due to
  # the hourly sampling set this first moored observation to the
    ↪ destination where the ship is
  # currently instead of the next destination.
655 i <- 1
  ERP_AIS_relevant$destination_temp <- NA
  while (i < nrow(ERP_AIS_relevant)) {

    setTxtProgressBar(pb, i)
660 ERP_AIS_relevant$destination_temp[i] <-
      ↪ clean_destination(ERP_AIS_relevant$destination[i])
    j <- 1

    while (i + j <= nrow(ERP_AIS_relevant) && ERP_AIS_relevant$imo[i]
      ↪ == ERP_AIS_relevant$imo[i + j] && ERP_AIS_relevant$route_id[i]
      ↪ == ERP_AIS_relevant$route_id[i + j] &&
      ↪ ERP_AIS_relevant$destination[i] ==
      ↪ ERP_AIS_relevant$destination[i + j]) {
      ERP_AIS_relevant$destination_temp[i + j] <-
        ↪ ERP_AIS_relevant$destination_temp[i]
665   j <- j + 1
    }

    if (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i + j] &&
      ↪ ERP_AIS_relevant$route_id[i] == ERP_AIS_relevant$route_id[i +
      ↪ j] && ERP_AIS_relevant$status[i + j] == 5 &&
      ↪ ERP_AIS_relevant$status[i + j - 1] != 5 &&
      ↪ ERP_AIS_relevant$destination[i + j] !=
      ↪ ERP_AIS_relevant$destination[i + j - 1]) {
```

```
      ERP_AIS_relevant$destination_temp[i + j] <-
        ↪ ERP_AIS_relevant$destination_temp[i + j - 1]
      j <- j + 1
    }
    i <- i + j
  }

  # Find port name based on Port code
  look_up_code <- function(ERP_AIS_relevant, letter, min, temp = NA,
    ↪ list = ports_list_code) {
    ports <- list[[letter]]

    for (j in 1:nrow(ports)) {
      if (is.na(temp))
        tmp <-stringdist(ERP_AIS_relevant$destination_temp,
          ↪ toupper(ports$PortCode[j]), method = "osa")
      else
        tmp <- stringdist(temp, toupper(ports$PortCode[j]), method =
          ↪ "osa")

      if (tmp < min) {
        if (tmp == 0) {
          ERP_AIS_relevant$destination_standard <-
            ↪ toupper(ports$PortName[j])
          ERP_AIS_relevant$destination_name <- NA
          ERP_AIS_relevant$destination_flag <- T
          break
        }

        else {
          min <- tmp
          ERP_AIS_relevant$destination_name <-
            ↪ toupper(ports$PortName[j])
        }
      }

      else if (tmp == min) {
        ERP_AIS_relevant$destination_name <-
          ↪ paste(ERP_AIS_relevant$destination_name,
          ↪ toupper(ports$PortName[j]), sep = ";")
      }
    }
    return(list(ERP_AIS_relevant, min))
  }
  # Find port name based on port name (remove typos)
  look_up_name <- function(ERP_AIS_relevant, letter, min, temp = NA,
    ↪ list = ports_list_name) {
    ports <- list[[letter]]
    for (j in ports) {
      if (is.na(temp))
        tmp <- stringdist(ERP_AIS_relevant$destination_temp,
          ↪ toupper(j), method = "osa")
      else
        tmp <- stringdist(temp, toupper(j), method = "osa")
      if (tmp < min) {
```

```
        if (tmp == 0) {
          ERP_AIS_relevant$destination_standard <- toupper(j)
          ERP_AIS_relevant$destination_name <- NA
          ERP_AIS_relevant$destination_flag <- T
          break
        }

        else {
          min <- tmp
          ERP_AIS_relevant$destination_name <- toupper(j)
        }
      }

      else if (tmp == min) {
        ERP_AIS_relevant$destination_name <-
          ↪ paste(ERP_AIS_relevant$destination_name, toupper(j), sep =
          ↪ ";")
      }
    }
    return(list(ERP_AIS_relevant, min))
  }
  # Set the destination so only names are used and typos etc are
    ↪ removed
  std_destination <- function(ERP_AIS_relevant, ports, WPI) {
    pb <- txtProgressBar( min = 0, max = nrow(ERP_AIS_relevant), style
      ↪ = 3, title = "Standardising destinations")
    ERP_AIS_relevant$destination_standard <- NA
    ERP_AIS_relevant$destination_name <- NA
    ERP_AIS_relevant$destination_flag <- F

    library(stringdist)

    i <- 1
    while (i <= nrow(ERP_AIS_relevant)) {
      setTxtProgressBar(pb, i)

      if (ERP_AIS_relevant$status[i] == 5 &&
        ↪ !is.na(ERP_AIS_relevant$destination_temp[i])) {
        min <- 1000000000
        list <- look_up_name(ERP_AIS_relevant[i,],
          ↪ substring(ERP_AIS_relevant$destination_temp[i], 1, 1), min)
        ERP_AIS_relevant[i,] <- list[[1]]
        min <- list[[2]]

        if (!ERP_AIS_relevant$destination_flag[i]) {
          list <- look_up_code( ERP_AIS_relevant[i,],
            ↪ substring(ERP_AIS_relevant$destination_temp[i], 1, 1),
            ↪ min)
          ERP_AIS_relevant[i,] <- list[[1]]
          min <- list[[2]]
        }

        if (!grepl(";", ERP_AIS_relevant$destination_name[i]) &&
          ↪ !is.na(ERP_AIS_relevant$destination_name[i]) &&
          ↪ !ERP_AIS_relevant$destination_flag[i]) {
```

```
      ERP_AIS_relevant$destination_standard[i] <-
        ↪ ERP_AIS_relevant$destination_name[i]
      ERP_AIS_relevant$destination_flag[i] <- T
    }

    else if (grepl(".*\\s.+", ERP_AIS_relevant$destination_temp[i])
      ↪ && !ERP_AIS_relevant$destination_flag[i]) {
      temp <- ERP_AIS_relevant$destination_temp[i]

      while (grepl(".*\\s.+", temp)) {
        temp <- sub("\\s[[:alpha:]]*$", "", temp)
        list <- look_up_name( ERP_AIS_relevant[i,],
          ↪ substring(ERP_AIS_relevant$destination_temp[i], 1, 1),
          ↪ min, temp )
        ERP_AIS_relevant[i,] <- list[[1]]
        min <- list[[2]]

        if (ERP_AIS_relevant$destination_flag[i])
          break

        if (!ERP_AIS_relevant$destination_flag[i]) {
          list <- look_up_code( ERP_AIS_relevant[i,],
            ↪ substring(ERP_AIS_relevant$destination_temp[i], 1,
            ↪ 1), min, temp)
          ERP_AIS_relevant[i,] <- list[[1]]
          min <- list[[2]]
        }

        if (ERP_AIS_relevant$destination_flag[i])
          break

        if (!grepl(";", ERP_AIS_relevant$destination_name[i]) &&
          ↪ !is.na(ERP_AIS_relevant$destination_name[i]) &&
          ↪ !ERP_AIS_relevant$destination_flag[i]) {
          ERP_AIS_relevant$destination_standard[i] <-
            ↪ ERP_AIS_relevant$destination_name[i]
          ERP_AIS_relevant$destination_flag[i] <- T

        }
      }
    }

    if (grepl("^.+\\s", ERP_AIS_relevant$destination_temp[i]) &&
        !ERP_AIS_relevant$destination_flag[i]) {
      temp <- ERP_AIS_relevant$destination_temp[i]

      while (grepl("^.+\\s", temp)) {
        temp <- sub("^[[:alpha:]]+\\s", "", temp)
        list <- look_up_name( ERP_AIS_relevant[i,],
          ↪ substring(ERP_AIS_relevant$destination_temp[i], 1, 1),
          ↪ min, temp)
        ERP_AIS_relevant[i,] <- list[[1]]
        min <- list[[2]]

        if (ERP_AIS_relevant$destination_flag[i])
```

```
            break

805         if (!ERP_AIS_relevant$destination_flag[i]) {
            list <- look_up_code(ERP_AIS_relevant[i,],
              ↪ substring(ERP_AIS_relevant$destination_temp[i], 1,
              ↪ 1), min, temp)
            ERP_AIS_relevant[i,] <- list[[1]]
            min <- list[[2]]
          }
810
          if (ERP_AIS_relevant$destination_flag[i])
            break

          if (!grepl(";", ERP_AIS_relevant$destination_name[i]) &&
815           !is.na(ERP_AIS_relevant$destination_name[i]) &&
              !ERP_AIS_relevant$destination_flag[i]) {
            ERP_AIS_relevant$destination_standard[i] <-
              ↪ ERP_AIS_relevant$destination_name[i]
            ERP_AIS_relevant$destination_flag[i] <- T

820       }
        }
      }

      k <- 1
825   while (i - k > 0 && ERP_AIS_relevant$status[i - k] != 5 &&
        ↪ ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - k] &&
        ↪ ERP_AIS_relevant$route_id[i] == ERP_AIS_relevant$route_id[i
        ↪ - k]) {
        if (ERP_AIS_relevant$destination_flag[i]) {
          ERP_AIS_relevant$destination_standard[i - k] <-
            ↪ ERP_AIS_relevant$destination_standard[i]
          ERP_AIS_relevant$destination_flag[i - k] <- T
        }
830
        else
          ERP_AIS_relevant$destination_name[i - k] <-
            ↪ ERP_AIS_relevant$destination_name[i]

        k <- k + 1
835   }

      k <- 1
      while (i + k <= nrow(ERP_AIS_relevant) &&
        ↪ ERP_AIS_relevant$status[i + k] == 5 &&
        ↪ ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i + k] &
        ↪ ERP_AIS_relevant$route_id[i] == ERP_AIS_relevant$route_id[i
        ↪ + k]) {
        if (ERP_AIS_relevant$destination_flag[i]) {
840       ERP_AIS_relevant$destination_standard[i + k] <-
            ↪ ERP_AIS_relevant$destination_standard[i]
          ERP_AIS_relevant$destination_flag[i + k] <- T
        }

        else
```

```
            ERP_AIS_relevant$destination_name[i + k]
              ↪ <-ERP_AIS_relevant$destination_name[i]

            k <- k + 1
          }
          i <- i + k
        }

        else
          i <- i + 1
      }


      ERP_AIS_relevant$destination_knn <- NA

      library(class)
      train <- WPI[, 15:16]
      test <- ERP_AIS_relevant[ERP_AIS_relevant$status == 5, 5:6]
      train_label <- as.factor(WPI$Main_port_name)

      dest <- knn(train = train, test = test, cl = train_label, k = 1)
      test$dest <- as.character(dest)

      i <- 1
      j <- 1
      while (i < nrow(ERP_AIS_relevant)) {
        setTxtProgressBar(pb, i)
        if (ERP_AIS_relevant$latitude[i] == test$latitude[j] &&
          ↪ ERP_AIS_relevant$longitude[i] == test$longitude[j]) {
          if (!is.na(ERP_AIS_relevant$destination_name[i])  &&
            ↪ !is.na(ERP_AIS_relevant$destination_temp[i])) {
            ERP_AIS_relevant$destination_knn[i] <- toupper(test$dest[j])
            k <- 1
            while (i - k > 0 && ERP_AIS_relevant$status[i - k] != 5) {
              ERP_AIS_relevant$destination_knn[i - k] <-
                ↪ toupper(test$dest[j])
              k <- k + 1
            }

          }
          j <- j + 1
        }
        i <- i + 1
      }
      return(ERP_AIS_relevant)
    }
    ERP_AIS_relevant <- std_destination(ERP_AIS_relevant, ports, WPI)

    # Retrieve shipping lines from the ship name
    shipping_lines <- function(ERP_AIS_relevant) {
      pb <-
        txtProgressBar(
          min = 0,
          max = nrow(ERP_AIS_relevant),
          style = 3,
```

```
    title = "Retrieving shipping lines"
  )
ERP_AIS_relevant$shipping_line <- "UNKOWN"
i <- 1
while (i <= nrow(ERP_AIS_relevant)) {
  setTxtProgressBar(pb, i)
  if (grepl("^AL\\s", ERP_AIS_relevant$name[i])) {
    ERP_AIS_relevant$shipping_line[i] <- "AL"
    i <- i + 1
    while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
      ↪ &&
           i <= nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$shipping_line[i] <- "AL"
      i <- i + 1
    }
  }

  else if (grepl("^APL\\s", ERP_AIS_relevant$name[i])) {
    ERP_AIS_relevant$shipping_line[i] <- "APL"
    i <- i + 1
    while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
      ↪ &&
           i <= nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$shipping_line[i] <- "APL"
      i <- i + 1
    }
  }

  else if (grepl("^ASTRO\\s", ERP_AIS_relevant$name[i])) {
    ERP_AIS_relevant$shipping_line[i] <- "ASTRO"
    i <- i + 1
    while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
      ↪ &&
           i <= nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$shipping_line[i] <- "ASTRO"
      i <- i + 1
    }
  }

  else if (grepl("^ATLANTIC\\s", ERP_AIS_relevant$name[i])) {
    ERP_AIS_relevant$shipping_line[i] <- "ATLANTIC"
    i <- i + 1
    while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
      ↪ &&
           i <= nrow(ERP_AIS_relevant)) {
      ERP_AIS_relevant$shipping_line[i] <- "ATLANTIC"
      i <- i + 1
    }
  }

  else if (grepl("^BBC\\s", ERP_AIS_relevant$name[i])) {
    ERP_AIS_relevant$shipping_line[i] <- "BBC"
    i <- i + 1
    while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
      ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BBC"
          i <- i + 1
        }
      }

      else if (grepl("^BF\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "BF"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BF"
          i <- i + 1
        }
      }

      else if (grepl("^BG\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "BG"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BG"
          i <- i + 1
        }
      }

      else if (grepl("^BOMAR\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "BOMAR"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BOMAR"
          i <- i + 1
        }
      }

      else if (grepl("^BOX\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "BOX"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BOX"
          i <- i + 1
        }
      }

      else if (grepl("^CAP\\sSAN\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CAP SAN"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CAP SAN"
          i <- i + 1
        }
      }

      else if (grepl("^CMA\\sCGM\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CMA CGM"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CMA CGM"
          i <- i + 1
        }
      }

      else if (grepl("^CMACGM\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CMA CGM"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CMA CGM"
          i <- i + 1
        }
      }

      else if (grepl("^CONMAR\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CONMAR"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CONMAR"
          i <- i + 1
        }
      }

      else if (grepl("^CONTI\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CONTI"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CONTI"
          i <- i + 1
        }
      }

      else if (grepl("^COSCO\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "COSCO"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "COSCO"
          i <- i + 1
        }
      }

      else if (grepl("^CSCL\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "CSCL"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "CSCL"
          i <- i + 1
        }
      }

      else if (grepl("\\sEXPRESS$", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "EXPRESS"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "EXPRESS"
          i <- i + 1
        }
      }

      else if (grepl("^DS\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "DS"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "DS"
          i <- i + 1
        }
      }

      else if (grepl("^E\\.\\sR\\.", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "E.R."
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "E.R."
          i <- i + 1
        }
      }

      else if (grepl("^E\\.R\\.\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "E.R."
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "E.R."
          i <- i + 1
        }
      }

      else if (grepl("^ECL\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "ECL"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "ECL"
          i <- i + 1
        }
      }

      else if (grepl("^EM\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "EM"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "EM"
          i <- i + 1
        }
      }

      else if (grepl("^EVER\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "EVERGREEN"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "EVERGREEN"
          i <- i + 1
        }
      }

      else if (grepl("^FRISIA\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "FRISIA"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "FRISIA"
          i <- i + 1
        }
      }

      else if (grepl("\\sBRIDGE$", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "BRIDGE"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "BRIDGE"
          i <- i + 1
        }
      }

      else if (grepl("^HANJIN\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "HANJIN"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "HANJIN"
          i <- i + 1
        }
      }

      else if (grepl("^HANSA\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "HANSA"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "HANSA"
          i <- i + 1
        }
      }

      else if (grepl("^HS\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "HS"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "HS"
          i <- i + 1
        }
      }

      else if (grepl("^HYUNDAI\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "HYUNDAI"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "HYUNDAI"
          i <- i + 1
        }
      }

      else if (grepl("^ICE\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "ICE"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "ICE"
          i <- i + 1
        }
      }

      else if (grepl("^JORK\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "JORK"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "JORK"
          i <- i + 1
        }
      }

      else if (grepl("^JPO\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "JPO"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "JPO"
          i <- i + 1
        }
      }

      else if (grepl("\\sSCHEPERS$", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "SCHEPERS"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "SCHEPERS"
          i <- i + 1
        }
      }

      else if (grepl("MAERSK", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "MAERSK"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "MAERSK"
          i <- i + 1
        }
      }

      else if (grepl("^MAX\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "MAX"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "MAX"
          i <- i + 1
        }
      }

      else if (grepl("^MOL\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "MOL"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "MOL"
          i <- i + 1
        }
      }

      else if (grepl("^MSC\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "MSC"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "MSC"
          i <- i + 1
        }
      }

      else if (grepl("^MV\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "MV"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "MV"
          i <- i + 1
        }
      }

      else if (grepl("^NORDIC\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "NORDIC"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "NORDIC"
          i <- i + 1
        }
      }

      else if (grepl("^NYK\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "NYK"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "NYK"
          i <- i + 1
        }
      }

      else if (grepl("^OOCL\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "OOCL"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "OOCL"
          i <- i + 1
        }
      }

      else if (grepl("^OPDR\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "OPDR"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "OPDR"
          i <- i + 1
        }
      }

      else if (grepl("^PHOENIX\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "PHOENIX"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "PHOENIX"
          i <- i + 1
        }
      }

      else if (grepl("^THALASSA\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "THALASSA"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "THALASSA"
          i <- i + 1
        }
      }

      else if (grepl("^WES\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "WES"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
```

```
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "WES"
          i <- i + 1
        }
      }

      else if (grepl("^WILSON\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "WILSON"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "WILSON"
          i <- i + 1
        }
      }

      else if (grepl("^XIN\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "XIN"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "XIN"
          i <- i + 1
        }
      }

      else if (grepl("^YM\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "YM"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "YM"
          i <- i + 1
        }
      }

      else if (grepl("^ZIM\\s", ERP_AIS_relevant$name[i])) {
        ERP_AIS_relevant$shipping_line[i] <- "ZIM"
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
          ERP_AIS_relevant$shipping_line[i] <- "ZIM"
          i <- i + 1
        }
      }

      else {
        i <- i + 1
        while (ERP_AIS_relevant$imo[i] == ERP_AIS_relevant$imo[i - 1]
          ↪ &&
            i <= nrow(ERP_AIS_relevant)) {
```

```
            i <- i + 1
          }
        }
      }
      return(ERP_AIS_relevant)
}
ERP_AIS_relevant <- shipping_lines(ERP_AIS_relevant)

ERP_AIS_Backup3 <- ERP_AIS_relevant

# Create a dataframe that contains for every combination of imo and
  ↪ route_id the destinations.
routes <- ERP_AIS_relevant %>%
  group_by(imo, route_id) %>%
  distinct(destination_temp, destination_standard, destination_name,
    ↪ destination_knn)

# NOW WE NEED TO DO SOME MANUAL WORK, THIS WILL TAKE YOU A DAY AT
  ↪ LEAST.
# I ADVISE TO REPEAT THIS STEP AT LEAST TWO TIMES TO BE SURE YOU HAVE
  ↪ NOT MISSED ANYTHING

# In the dataframe check the routes. Every route should start with a
  ↪ standardized destination
# that is not Rotterdam and finish in a standardized destination that
  ↪ is the port of Rotterdam.
# If the route starts with Rotterdam the captain forgot to change the
  ↪ location (also possible
# with other destinations but at the first run we can not check
  ↪ this.)
# If routes are very short, max 2 destinations, check that specific
  ↪ route in the dataset.
# Ships may be laying at anchor for almost the entire route and sail
  ↪ into rotterdam to pick up
# new containers for example.
# Furthermore check for wrongly standardized destination
# Create an excel file "dest_rename.xls" call the first column "IMO",
  ↪ the second "ROUTE",
# the third "Wrong", the fourth "Right", the fith "Latitude" and the
  ↪ sixth "Longitude.
# Put the IMO and Route ID in the corresponding column if we want to
  ↪ change something.
# In column "Wrong" put the wrong destination and in "Right" the
  ↪ correct destination.
# When the captain forgets to change a destination we need to do this
  ↪ manualy for the
# first observation where the ship is moored in the port. Set the
  ↪ latitude and longitude
# of that observation in the corresponding columns.
# When we want to remove a specific route set the IMO and route in
  ↪ the corresponding column
# and put "REMOVE" in column "WRONG". When we want to delete all the
  ↪ journeys of a vessel
# only set the IMO.
# Make sure you add things you want to change in the order you
  ↪ encounter them. This makes the
```

```
# processing easier.

# Read the excel file just created.
library(readxl)

manual_rename <- read_excel("dest_rename.xlsx")
View(manual_rename)

# First this function creates a dataframe with routes to remove and then removes these from
# the dataset. Then it changes the destination you want to change based on the excel file.
# Because this excel file is created manually we output the needed info from the dataset and
# the row from the excel file to check if the right rows in the dataset are changed. Furthermore
# the order in which changes are made should be the same as in the excel file. So when a row
# is skipped note this row and after execution check why it was skipped. Probably there is a
# mistake in a column in the excel file.
# At the end of the function the destinations are again standardized.
# When the function finishes change the excel file for rows that were not changes and also
# repeat the manual check to be sure nothing is missed and add thing you missed to the
# excel file.
manual_changes <- function(ERP_AIS_relevant, manual_rename) {
  remove <- data.frame(imo = numeric(0), route_id = numeric(0))
  for (i in 1:nrow(manual_rename)) {
    if (grepl("REMOVE", manual_rename$Wrong[i])) {
      remove <- bind_rows(remove, data.frame(imo = manual_rename$IMO[i], route_id = manual_rename$ROUTE[i]))
    }
  }

  j <- 1
  pb <- txtProgressBar(min = 0, max = nrow(remove), style = 3, title = "Construct routes")
  for (i in 1:nrow(remove)) {
    setTxtProgressBar(pb, i)

    if (!is.na(remove$route_id[i])) {
      while (!(ERP_AIS_relevant$imo[j] == remove$imo[i] &&
        ERP_AIS_relevant$route_id[j] == remove$route_id[i]) && j <=
        nrow(ERP_AIS_relevant))
        j <- j + 1

      while (ERP_AIS_relevant$imo[j] == remove$imo[i] &&
        ERP_AIS_relevant$route_id[j] == remove$route_id[i] && j <=
        nrow(ERP_AIS_relevant)) {
        ERP_AIS_relevant$imo[j] <- NA
        j <- j + 1
      }
    }
```

```
    else{
      while (ERP_AIS_relevant$imo[j] != remove$imo[i] && j <=
        ↪ nrow(ERP_AIS_relevant))
        j <- j + 1

      while (ERP_AIS_relevant$imo[j] == remove$imo[i] && j <=
        ↪ nrow(ERP_AIS_relevant)) {
        ERP_AIS_relevant$imo[j] <- NA
        j <- j + 1
      }
    }

  }
  ERP_AIS_relevant <- ERP_AIS_relevant[!is.na(ERP_AIS_relevant$imo),
    ↪ ]

  k <- 0
  wrong <- unique(manual_rename$Wrong)
  for (i in 1:nrow(ERP_AIS_relevant)) {
    if (ERP_AIS_relevant$destination_temp[i] %in% wrong) {
      for (j in 1:nrow(manual_rename))

        if (ERP_AIS_relevant$imo[i] == manual_rename$IMO[j] &&
          ↪ ERP_AIS_relevant$route_id[i] == manual_rename$ROUTE[j] &&
          ↪ ERP_AIS_relevant$destination_temp[i] ==
          ↪ manual_rename$Wrong[j] && !is.na(manual_rename$Wrong[j])
          ↪ && !is.na(ERP_AIS_relevant$destination_temp[i])) {
          if (k != j) {
            print(ERP_AIS_relevant$imo[i])
            print(ERP_AIS_relevant$route_id[i])
            print(ERP_AIS_relevant$destination_temp[i])
            print(ERP_AIS_relevant$latitude[i])
            print(ERP_AIS_relevant$longitude[i])
            print(manual_rename[j, ])
            Sys.sleep(2)
            k <- j
          }
          if (is.na(manual_rename$latitude[i])) {
            ERP_AIS_relevant$destination_temp[i] <-
              ↪ manual_rename$Right[j]
            break
          }

          else if (ERP_AIS_relevant$latitude[i] ==
            ↪ manual_rename$latitude[j] &&
            ↪ ERP_AIS_relevant$longitude[i] ==
            ↪ manual_rename$longitude[j]) {
            ERP_AIS_relevant$destination_temp[i] <-
              ↪ manual_rename$Right[j]
            break
          }
        }
      else if(ERP_AIS_relevant$imo[i] == manual_rename$IMO[j] &&
        ↪ ERP_AIS_relevant$route_id[i] == manual_rename$ROUTE[j] &&
        ↪ is.na(ERP_AIS_relevant$destination_temp[i]) &&
        ↪ is.na(manual_rename$Wrong[j]))
```

```
            ERP_AIS_relevant$destination_temp[i] <-
              ↪ manual_rename$Right[j]
        }
      }

1520    ERP_AIS_relevant <- std_destination(ERP_AIS_relevant, ports, WPI)
      return(ERP_AIS_relevant)
    }
    ERP_AIS_relevant <- manual_changes(ERP_AIS_relevant, manual_rename)

1525 ERP_AIS_Backup4 <- ERP_AIS_relevant
    ERP_AIS_useful <- ERP_AIS_relevant

    ######### Dimension reduction #########
    summary(ERP_AIS_useful)
1530 #Remove attributes that are not thrustworty(MMSI) or do not provide
      ↪ additional information(e.g. type)
    ERP_AIS_useful$mmsi <- NULL
    ERP_AIS_useful$heading <- NULL
    ERP_AIS_useful$callsign <- NULL
    ERP_AIS_useful$type <- NULL
1535 ERP_AIS_useful$bow <- NULL
    ERP_AIS_useful$stern <- NULL
    ERP_AIS_useful$port <- NULL
    ERP_AIS_useful$starboard <- NULL
    ERP_AIS_useful$destination_temp <- NULL
1540 ERP_AIS_useful$destination_name <- NULL
    ERP_AIS_useful$destination_knn <- NULL
    ERP_AIS_useful$destination_flag <- NULL
    ERP_AIS_useful$destination <- NULL
    ERP_AIS_useful$name <- NULL
1545
    ERP_AIS_useful$destination_standard[154796:154989] <- "FELIXSTOWE"

    for(i in 1:nrow(ERP_AIS_useful)){
      if(ERP_AIS_useful$imo[i] == 9454450 && ERP_AIS_useful$status[i] ==
        ↪ 15)
1550    ERP_AIS_useful$status[i] <- 5
      else if(is.na(ERP_AIS_useful$width[i])){
        ERP_AIS_useful$width[i] <- 40
        ERP_AIS_useful$length[i] <- 299.9
      }
1555  else if(is.na(ERP_AIS_useful$destination_standard[i]))
        ERP_AIS_useful$destination_standard[i] <- "IZMIR"

      # Remove special case, long waiting times due to christmas and NYE
      else if (ERP_AIS_useful$imo[i] == 9178537 &&
        ↪ ERP_AIS_useful$route_id[i]==7)
1560    ERP_AIS_useful$imo[i] <- NA

      # Remove special case, long at anchor
      else if (ERP_AIS_useful$imo[i] == 9525883 &&
        ↪ ERP_AIS_useful$route_id[i]==11)
        ERP_AIS_useful$imo[i] <- NA
1565
```

```
    # Remove special cases, moored exceptionally long in Moerdijk
    else if (ERP_AIS_useful$imo[i] == 9162679 &&
      ↪ ERP_AIS_useful$route_id[i]==35)
      ERP_AIS_useful$imo[i] <- NA

    else if (ERP_AIS_useful$imo[i] == 9318931 &&
      ↪ ERP_AIS_useful$route_id[i]==56)
      ERP_AIS_useful$imo[i] <- NA

    else if (ERP_AIS_useful$imo[i] == 9318931 &&
      ↪ ERP_AIS_useful$route_id[i]==59)
      ERP_AIS_useful$imo[i] <- NA

    else if (ERP_AIS_useful$imo[i] == 9318931 &&
      ↪ ERP_AIS_useful$route_id[i]==67)
      ERP_AIS_useful$imo[i] <- NA
  }
  ERP_AIS_useful <- ERP_AIS_useful[!is.na(ERP_AIS_useful$imo),]

  ERP_AIS_useful$status <- factor(ERP_AIS_useful$status, ordered=F)
  ERP_AIS_useful$shipping_line <- factor(ERP_AIS_useful$shipping_line,
    ↪ ordered = F)

  library(dummies)
  ERP_AIS_useful <- cbind(ERP_AIS_useful, dummy(ERP_AIS_useful$status,
    ↪ sep="_status"))
  ERP_AIS_useful <- cbind(ERP_AIS_useful,
    ↪ dummy(ERP_AIS_useful$shipping_line, sep="_shipping_line_"))

  ######### Route identification############

  route_construction <- function(ERP_AIS_useful){
    i <- 2
    j <- 1
    routes <-data.frame(imo = numeric(0), route_id = numeric(0), route
      ↪ = character(0))

    pb <- txtProgressBar(min = 0, max = nrow(ERP_AIS_useful), style =
      ↪ 3, title = "Construct routes")

    while (ERP_AIS_useful$imo[i] == ERP_AIS_useful$imo[i - 1] && i <=
      ↪ nrow(ERP_AIS_useful)) {
      setTxtProgressBar(pb, i)

      routes <-bind_rows(routes, data.frame(imo =
        ↪ ERP_AIS_useful$imo[i], route_id =
        ↪ ERP_AIS_useful$route_id[i],route = NA))

      while (ERP_AIS_useful$route_id[i] == routes$route_id[j] && i <=
        ↪ nrow(ERP_AIS_useful)) {

        if (is.na(routes$route[j])) {
          routes$route[j] <-
            as.character(ERP_AIS_useful$destination_standard[i])
        }
```

```
        else {
          routes$route[j] <-
            paste(routes$route[j],
              ↪ ERP_AIS_useful$destination_standard[i], sep = " -> ")
        }

        i <- i + 1
        while (ERP_AIS_useful$destination_standard[i] ==
          ↪ ERP_AIS_useful$destination_standard[i - 1] && i <=
          ↪ nrow(ERP_AIS_useful))
          i <- i + 1
      }
      i <- i + 1
      j <- j + 1
    }
    return(routes)
}
routes <- route_construction(ERP_AIS_useful)

route_count <- function(routes){
  route_class <- list(data.frame(route = character(0), times_taken =
    ↪ numeric(0)), imo = vector("list",
    ↪ length(unique(routes$route))))

  names(route_class$imo) <- unique(routes$route)

  pb <-txtProgressBar(min = 0, max = length(unique(routes$route)),
    ↪ style = 3, title = "Create dataframe with unique routes")
  j<-0

  for (i in unique(routes$route)) {
    j <- j+1
    setTxtProgressBar(pb, j)
    route_class[[1]] <- bind_rows(route_class[[1]], data.frame(route
      ↪ = i, times_taken = 0))
  }

  route_class[[1]]$route <- as.character(route_class[[1]]$route)
  pb <- txtProgressBar(min = 0, max = nrow(route_class[[1]]), style =
    ↪ 3, title = "count routes")

  for (i in 1:nrow(route_class[[1]])) {
    setTxtProgressBar(pb, i)
    for (j in 1:nrow(routes)) {
      if (route_class[[1]]$route[i] == routes$route[j]) {
        route_class[[1]]$times_taken[i] <-
          ↪ route_class[[1]]$times_taken[i] + 1

        # if(is.null(route_class$imo[[routes$route[j]]]))
        #   route_class$imo[[routes$route[j]]] <- routes$imo[j]
        #
        # else if(routes$imo[j] %in%
          ↪ route_class$imo[[routes$route[j]]])
        #   next
```

```
            #
            # else
            #   route_class$imo[[routes$route[j]]] <-
            #     c(route_class$imo[[routes$route[j]]], routes$imo[j])
          }
        }
      }
      route_class[[1]] <- arrange(route_class[[1]], desc(times_taken))

      return(route_class)
    }
    route_class <- route_count(routes)

    useful_routes <- data.frame(imo = numeric(0), route_id = numeric(0),
      ↪ route = character(0))

    for(i in 1:nrow(routes)){
      if(routes$route[i] %in% route_class[[1]]$route[1:6])
        useful_routes <- bind_rows(useful_routes, data.frame(imo =
          ↪ routes$imo[i], route_id = routes$route_id[i], route =
          ↪ routes$route[i]))
    }
    useful_routes$route <- as.character(useful_routes$route)

    AIS_data <- semi_join(ERP_AIS_useful, useful_routes)
    AIS_data <- arrange(AIS_data, timestamp)

    vessels <- data.frame(imo = numeric(0), route = character(0))
    identified <- data.frame(imo = numeric(0), time = numeric(0), voyage
      ↪ = numeric(0), pred_bool = logical())
    for(i in 1:nrow(AIS_data)){
      pred_imo <- F
      if(AIS_data$imo[i] %in% vessels$imo){
        for(j in 1:nrow(vessels)){

          if(AIS_data$imo[i] == vessels$imo[j] && AIS_data$latitude[i] >
            ↪ 51.85 && AIS_data$latitude[i] < 52.0 &&
            ↪ AIS_data$longitude[i] > 3.95 && AIS_data$longitude[i] < 4.6
            ↪ && AIS_data$status[i] == 5){
            break
          }

          else if(AIS_data$imo[i] == vessels$imo[j] &&
            ↪ AIS_data$destination_standard[i] ==
            ↪ substring(vessels$route[j], nchar(vessels$route[j]) -
            ↪ nchar(AIS_data$destination_standard[i])+1,
            ↪ nchar(vessels$route[j]))){
            break
          }

          else if(AIS_data$imo[i] == vessels$imo[j] &&
            ↪ AIS_data$destination_standard[i] !=
            ↪ substring(vessels$route[j], nchar(vessels$route[j]) -
            ↪ nchar(AIS_data$destination_standard[i]),
            ↪ nchar(vessels$route[j]))){
```

```
            vessels$route[j] <- paste(vessels$route[j],
              ↪ AIS_data$destination_standard[i], sep=" -> ")
            break
          }
        }
      }

      else{

        vessels <- bind_rows(vessels, data.frame(imo = AIS_data$imo[i],
          ↪ route = AIS_data$destination_standard[i]))
        j <- nrow(vessels)
      }

      for(k in 1:6){
        if(vessels$route[j] ==
          ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))
          ↪ && AIS_data$imo[i] %in%
          ↪ route_class[[2]][[route_class[[1]]$route[k]]]){
          print(paste("Possible route for vessel", AIS_data$imo[i],
            ↪ "during voyage", AIS_data$route_id[i], "based on IMO
            ↪ number, current route and destination is:"))
          print(route_class[[1]]$route[k])
          pred_imo <- T
        }
      }
      if(!pred_imo){

        for(k in 1:6){
          if(vessels$route[j] ==
            ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))){
            print(paste("Possible route for vessel", AIS_data$imo[i],
              ↪ "during voyage", AIS_data$route_id[i], "based on current
              ↪ route and destination is:"))
            print(route_class[[1]]$route[k])
          }
        }
      }

      identified <- bind_rows(identified, data.frame(imo =
        ↪ AIS_data$imo[i], time = AIS_data$timestamp[i], voyage =
        ↪ AIS_data$route_id[i], pred_bool = pred_imo))

      if(AIS_data$latitude[i] > 51.85 && AIS_data$latitude[i] < 52.0 &&
        ↪ AIS_data$longitude[i] > 3.95 && AIS_data$longitude[i] < 4.6 &&
        ↪ AIS_data$status[i] == 5){
        if(!pred_imo)
          route_class[[2]][[vessels$route[j]]] <-
            ↪ c(route_class[[2]][[vessels$route[j]]],vessels$imo[j])
        vessels <- vessels[-j,]
      }

    }
    changed <- 0
    identified <- identified[order(identified$imo, identified$voyage,
      ↪ identified$time),]
```

```
for(i in 2:nrow(identified)){
  if(identified$imo[i] == identified$imo[i-1] && identified$voyage[i]
    ↪ == identified$voyage[i-1] && identified$pred_bool[i] !=
    ↪ identified$pred_bool[i-1])
    changed <- changed+1
}
print(paste0("Accuracy is: ",(1-changed/nrow(useful_routes))*100,"%"
  ↪ ))

####### Predictions #####

route_construction <- function(ERP_AIS_useful){
  i <- 2
  j <- 1
  routes <-data.frame(imo = numeric(0), route_id = numeric(0), route
    ↪ = character(0))

  pb <- txtProgressBar(min = 0, max = nrow(ERP_AIS_useful), style =
    ↪ 3, title = "Construct routes")

  while (ERP_AIS_useful$imo[i] == ERP_AIS_useful$imo[i - 1] && i <=
    ↪ nrow(ERP_AIS_useful)) {
    setTxtProgressBar(pb, i)

    routes <-bind_rows(routes, data.frame(imo =
      ↪ ERP_AIS_useful$imo[i], route_id =
      ↪ ERP_AIS_useful$route_id[i],route = NA))

    while (ERP_AIS_useful$route_id[i] == routes$route_id[j] && i <=
      ↪ nrow(ERP_AIS_useful)) {

      if (is.na(routes$route[j])) {
        routes$route[j] <-
          as.character(ERP_AIS_useful$destination_standard[i])
      }

      else {
        routes$route[j] <-
          paste(routes$route[j],
            ↪ ERP_AIS_useful$destination_standard[i], sep = " -> ")
      }

      i <- i + 1
      while (ERP_AIS_useful$destination_standard[i] ==
        ↪ ERP_AIS_useful$destination_standard[i - 1] && i <=
        ↪ nrow(ERP_AIS_useful))
        i <- i + 1
    }
    i <- i + 1
    j <- j + 1
  }
  return(routes)
}
routes <- route_construction(ERP_AIS_useful)
```

```
route_count <- function(routes){
  route_class <- list(data.frame(route = character(0), times_taken =
    ↪ numeric(0)), imo = vector("list",
    ↪ length(unique(routes$route))))

  names(route_class$imo) <- unique(routes$route)

  pb <-txtProgressBar(min = 0, max = length(unique(routes$route)),
    ↪ style = 3, title = "Create dataframe with unique routes")
  j<-0

  for (i in unique(routes$route)) {
    j <- j+1
    setTxtProgressBar(pb, j)
    route_class[[1]] <- bind_rows(route_class[[1]], data.frame(route
      ↪ = i, times_taken = 0))
  }

  route_class[[1]]$route <- as.character(route_class[[1]]$route)
  pb <- txtProgressBar(min = 0, max = nrow(route_class[[1]]), style =
    ↪ 3, title = "count routes")

  for (i in 1:nrow(route_class[[1]])) {
    setTxtProgressBar(pb, i)
    for (j in 1:nrow(routes)) {
      if (route_class[[1]]$route[i] == routes$route[j]) {
        route_class[[1]]$times_taken[i] <-
          ↪ route_class[[1]]$times_taken[i] + 1

        # if(is.null(route_class$imo[[routes$route[j]]]))
        #   route_class$imo[[routes$route[j]]] <- routes$imo[j]
        #
        # else if(routes$imo[j] %in%
          ↪ route_class$imo[[routes$route[j]]])
        #   next
        #
        # else
        #   route_class$imo[[routes$route[j]]] <-
        #     c(route_class$imo[[routes$route[j]]], routes$imo[j])
      }
    }
  }
  route_class[[1]] <- arrange(route_class[[1]], desc(times_taken))

  return(route_class)
}
route_class <- route_count(routes)

useful_routes <- data.frame(imo = numeric(0), route_id = numeric(0),
  ↪ route = character(0))

for(i in 1:nrow(routes)){
  if(routes$route[i] %in% route_class[[1]]$route[1:6])
    useful_routes <- bind_rows(useful_routes, data.frame(imo =
      ↪ routes$imo[i], route_id = routes$route_id[i], route =
      ↪ routes$route[i]))
```

```
}
useful_routes$route <- as.character(useful_routes$route)

predictions <- data.frame(timestamp = numeric (0), imo = numeric(0),
  ↪ route_id = numeric(0), route = character(0), traveltime =
  ↪ numeric(0), est_traveltime = numeric(0), pred_traveltime =
  ↪ numeric(0), pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0),
  ↪ pred_imo = logical())
predictions_all <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsmin2 <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_allmin2 <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsplus2 <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_allplus2 <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictions_noETA <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_all_noETA <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsmin2_noETA <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_allmin2_noETA <- data.frame(timestamp = numeric (0), imo
  ↪ = numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsplus2_noETA <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
```

```
predictions_allplus2_noETA <- data.frame(timestamp = numeric (0), imo
  ↪ = numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictions_course <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_all_course <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsmin2_course <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_allmin2_course <- data.frame(timestamp = numeric (0), imo
  ↪ = numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
predictionsplus2_course <- data.frame(timestamp = numeric (0), imo =
  ↪ numeric(0), route_id = numeric(0), route = character(0),
  ↪ traveltime = numeric(0), est_traveltime = numeric(0),
  ↪ pred_traveltime = numeric(0), pred_ETA=numeric(0),ETA=numeric(0),
  ↪ ATA=numeric(0), pred_imo = logical())
predictions_allplus2_course <- data.frame(timestamp = numeric (0),
  ↪ imo = numeric(0), route_id = numeric(0), traveltime = numeric(0),
  ↪ est_traveltime = numeric(0), pred_traveltime = numeric(0),
  ↪ pred_ETA=numeric(0),ETA=numeric(0), ATA=numeric(0), pred_imo =
  ↪ logical())
neighbours <- numeric(6)

for(l in 1:6){
  neighbours[l] <- round(sqrt(route_class[[1]]$times_taken[l] - 1))
}

for(cross in 1:route_class[[1]]$times_taken[1]){
  route_class <- route_count(routes)
  cat("\n Step: ", cross, "\n")
  counter_route <- rep(1,6)


  train_routes <- data.frame(imo = numeric(0), route_id = numeric(0),
    ↪ route = character(0))
  test_routes <- data.frame(imo = numeric(0), route_id = numeric(0),
    ↪ route = character(0))
  for(i in 1:nrow(useful_routes)){
    for(j in 1:6){

      if(counter_route[j] == cross &&
        ↪ useful_routes$route[i]==route_class[[1]]$route[j]){
```

```
            test_routes <- bind_rows(test_routes, useful_routes[i,])
            counter_route[j] <- counter_route[j] +1
          }

          else if (useful_routes$route[i]==route_class[[1]]$route[j]){
            train_routes <- bind_rows(train_routes, useful_routes[i,])
            counter_route[j] <- counter_route[j] +1
          }
        }
      }

      train_sets <- vector("list", 6)
      names(train_sets) <- route_class[[1]]$route[1:6]

      train_set <- semi_join(ERP_AIS_useful, train_routes)
      train_data <- left_join(train_set, train_routes)
      pb <- txtProgressBar(min = 0, max = nrow(train_data), style = 3,
        ↪ title = "Construct training and test set")
      for(i in 1:nrow(train_data)){
        setTxtProgressBar(pb, i)
        train_sets[[train_data$route[i]]] <-
          ↪ bind_rows(train_sets[[train_data$route[i]]],
          ↪ train_data[i,1:65])
      }
      test_set <- semi_join(ERP_AIS_useful,test_routes)
      test_set <- arrange(test_set, timestamp)

      library(FNN)
      vessels <- data.frame(imo = numeric(0), route = character(0))
      for(i in 1:nrow(test_set)){
        pred_imo <- F
        if(test_set$imo[i] %in% vessels$imo){
          for(j in 1:nrow(vessels)){

            if(test_set$imo[i] == vessels$imo[j] &&
              ↪ test_set$destination_standard[i] ==
              ↪ substring(vessels$route[j], nchar(vessels$route[j]) -
              ↪ nchar(test_set$destination_standard[i])+1,
              ↪ nchar(vessels$route[j]))){
              break
            }

            else if(test_set$imo[i] == vessels$imo[j]){
              vessels$route[j] <- paste(vessels$route[j],
                ↪ test_set$destination_standard[i], sep=" -> ")
              break
            }
          }
        }

        else{
          vessels <- bind_rows(vessels, data.frame(imo = test_set$imo[i],
            ↪ route = test_set$destination_standard[i]))
          j <- nrow(vessels)
        }
```

```
        predictors <- c(4,5,7,13,17,18,22:65)
        for(k in 1:6){
          if(vessels$route[j] ==
            substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))
            && test_set$imo[i] %in%
            route_class[[2]][[route_class[[1]]$route[k]]]){
            pred <- knn.reg(train =
              train_sets[[route_class[[1]]$route[k]]][,predictors],
              test = test_set[i,predictors], y =
              as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              k = neighbours[k], algorithm="brute")
            predictions <- bind_rows(predictions, data.frame(timestamp =
              test_set$timestamp[i], imo = test_set$imo[i], route_id =
              test_set$route_id[i], route = route_class[[1]]$route[k],
              traveltime=round(test_set$traveltime[i]), est_traveltime
              = test_set$est_traveltime[i], pred_traveltime =
              pred$pred,
              pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred <- knn.reg(train =
              train_sets[[route_class[[1]]$route[k]]][,predictors],
              test = test_set[i,predictors], y =
              as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              k = neighbours[k]+2, algorithm="brute")
            predictionsplus2 <- bind_rows(predictionsplus2,
              data.frame(timestamp = test_set$timestamp[i], imo =
              test_set$imo[i], route_id = test_set$route_id[i], route =
              route_class[[1]]$route[k],
              traveltime=round(test_set$traveltime[i]), est_traveltime
              = test_set$est_traveltime[i], pred_traveltime =
              pred$pred,
              pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred <- knn.reg(train =
              train_sets[[route_class[[1]]$route[k]]][,predictors],
              test = test_set[i,predictors], y =
              as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              k = neighbours[k]-2, algorithm="brute")
            predictionsmin2 <- bind_rows(predictionsmin2,
              data.frame(timestamp = test_set$timestamp[i], imo =
              test_set$imo[i], route_id = test_set$route_id[i], route =
              route_class[[1]]$route[k],
              traveltime=round(test_set$traveltime[i]), est_traveltime
              = test_set$est_traveltime[i], pred_traveltime =
              pred$pred,
              pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred_imo <- T
          }
        }

        if(!pred_imo){
          for(k in 1:6){
            if(vessels$route[j] ==
              substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))){
```

```
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k], algorithm="brute")
            predictions <- bind_rows(predictions, data.frame(timestamp
              ↪ = test_set$timestamp[i], imo = test_set$imo[i],
              ↪ route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]+2, algorithm="brute")
            predictionsplus2 <- bind_rows(predictionsplus2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]-2, algorithm="brute")
            predictionsmin2 <- bind_rows(predictionsmin2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
          }
        }
      }

      predictors <- c(4:7,13,17,18,22:65)
      for(k in 1:6){
        if(vessels$route[j] ==
          ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))
          ↪ && test_set$imo[i] %in%
          ↪ route_class[[2]][[route_class[[1]]$route[k]]]){
          pred <- knn.reg(train =
            ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
            ↪ test = test_set[i,predictors], y =
            ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
            ↪ k = neighbours[k], algorithm="brute")
```

```
            predictions_course <- bind_rows(predictions,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
              ↪ = test_set$est_traveltime[i], pred_traveltime =
              ↪ pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]+2, algorithm="brute")
            predictionsplus2_course <- bind_rows(predictionsplus2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
              ↪ = test_set$est_traveltime[i], pred_traveltime =
              ↪ pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]-2, algorithm="brute")
            predictionsmin2_course <- bind_rows(predictionsmin2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
              ↪ = test_set$est_traveltime[i], pred_traveltime =
              ↪ pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred_imo <- T
          }
        }

        if(!pred_imo){
          for(k in 1:6){
            if(vessels$route[j] ==
              ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))){
              pred <- knn.reg(train =
                ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
                ↪ test = test_set[i,predictors], y =
                ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
                ↪ k = neighbours[k], algorithm="brute")
              predictions_course <- bind_rows(predictions,
                ↪ data.frame(timestamp = test_set$timestamp[i], imo =
                ↪ test_set$imo[i], route_id = test_set$route_id[i], route
                ↪ = route_class[[1]]$route[k],
                ↪ traveltime=round(test_set$traveltime[i]),
                ↪ est_traveltime = test_set$est_traveltime[i],
                ↪ pred_traveltime = pred$pred,
                ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
                ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
```

```
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]+2, algorithm="brute")
            predictionsplus2_course <- bind_rows(predictionsplus2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]-2, algorithm="brute")
            predictionsmin2_course <- bind_rows(predictionsmin2,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
          }
        }
      }

      predictors <- c(4,5,7,17,18,22:65)

      for(k in 1:6){
        if(vessels$route[j] ==
          ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))
          ↪ && test_set$imo[i] %in%
          ↪ route_class[[2]][[route_class[[1]]$route[k]]]){
          pred <- knn.reg(train =
            ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
            ↪ test = test_set[i,predictors], y =
            ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
            ↪ k = neighbours[k], algorithm="brute")
          predictions_noETA <- bind_rows(predictions_noETA,
            ↪ data.frame(timestamp = test_set$timestamp[i], imo =
            ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
            ↪ route_class[[1]]$route[k],
            ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
            ↪ = test_set$est_traveltime[i], pred_traveltime =
            ↪ pred$pred,
            ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
            ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
          pred <- knn.reg(train =
            ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
            ↪ test = test_set[i,predictors], y =
            ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
            ↪ k = neighbours[k]+2, algorithm="brute")
```

```
            predictionsplus2_noETA <- bind_rows(predictionsplus2_noETA,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
              ↪ = test_set$est_traveltime[i], pred_traveltime =
              ↪ pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]-2, algorithm="brute")
            predictionsmin2_noETA <- bind_rows(predictionsmin2_noETA,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route =
              ↪ route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]), est_traveltime
              ↪ = test_set$est_traveltime[i], pred_traveltime =
              ↪ pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=T))
            pred_imo <- T
            if(test_set$latitude[i] > 51.85 && test_set$latitude[i] <
              ↪ 52.0 && test_set$longitude[i] > 3.95 &&
              ↪ test_set$longitude[i] < 4.6 && test_set$status[i]==5){
              vessels <- vessels[-j,]
            }
          }
        }

        if(!pred_imo){
          for(k in 1:6){
            if(vessels$route[j] ==
              ↪ substring(route_class[[1]]$route[k],1,nchar(vessels$route[j]))){
              pred <- knn.reg(train =
                ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
                ↪ test = test_set[i,predictors], y =
                ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
                ↪ k = neighbours[k], algorithm="brute")
              predictions_noETA <- bind_rows(predictions_noETA,
                ↪ data.frame(timestamp = test_set$timestamp[i], imo =
                ↪ test_set$imo[i], route_id = test_set$route_id[i], route
                ↪ = route_class[[1]]$route[k],
                ↪ traveltime=round(test_set$traveltime[i]),
                ↪ est_traveltime = test_set$est_traveltime[i],
                ↪ pred_traveltime = pred$pred,
                ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
                ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
              pred <- knn.reg(train =
                ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
                ↪ test = test_set[i,predictors], y =
                ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
                ↪ k = neighbours[k]+2, algorithm="brute")
```

```
            predictionsplus2_noETA <- bind_rows(predictionsplus2_noETA,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
            pred <- knn.reg(train =
              ↪ train_sets[[route_class[[1]]$route[k]]][,predictors],
              ↪ test = test_set[i,predictors], y =
              ↪ as.data.frame(train_sets[[route_class[[1]]$route[k]]][,19]),
              ↪ k = neighbours[k]-2, algorithm="brute")
            predictionsmin2_noETA <- bind_rows(predictionsmin2_noETA,
              ↪ data.frame(timestamp = test_set$timestamp[i], imo =
              ↪ test_set$imo[i], route_id = test_set$route_id[i], route
              ↪ = route_class[[1]]$route[k],
              ↪ traveltime=round(test_set$traveltime[i]),
              ↪ est_traveltime = test_set$est_traveltime[i],
              ↪ pred_traveltime = pred$pred,
              ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600,ETA=
              ↪ test_set$eta[i], ATA= test_set$ATA[i], pred_imo=F))
          }
        }
        if(test_set$latitude[i] > 51.85 && test_set$latitude[i] < 52.0
          ↪ && test_set$longitude[i] > 3.95 && test_set$longitude[i] <
          ↪ 4.6 && test_set$status[i]==5){
          if(!(vessels$imo[j] %in%
            ↪ route_class[[2]][[vessels$route[j]]]))
            route_class[[2]][[vessels$route[j]]] <-
              ↪ c(route_class[[2]][[vessels$route[j]]],vessels$imo[j])
          vessels <- vessels[-j,]
        }
      }
    }

    predictors <- c(4,5,7,13,17,18,22:65)
    for(i in 1:nrow(test_set)){
      pred <- knn.reg(train = train_set[,predictors], test =
        ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
        ↪ = round(sqrt(nrow(train_set))), algorithm="brute")
      predictions_all <- bind_rows(predictions_all,
        ↪ data.frame(timestamp = test_set$timestamp[i], imo =
        ↪ test_set$imo[i], route_id = test_set$route_id[i],
        ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
        ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
        ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
        ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
      pred <- knn.reg(train = train_set[,predictors], test =
        ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
        ↪ = round(sqrt(nrow(train_set)))+2, algorithm="brute")
      predictions_allplus2 <- bind_rows(predictions_allplus2,
        ↪ data.frame(timestamp = test_set$timestamp[i], imo =
        ↪ test_set$imo[i], route_id = test_set$route_id[i],
        ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
        ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
        ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
        ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
```

```
    pred <- knn.reg(train = train_set[,predictors], test =
      ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
      ↪ = round(sqrt(nrow(train_set)))-2, algorithm="brute")
    predictions_allmin2 <- bind_rows(predictions_allmin2,
      ↪ data.frame(timestamp = test_set$timestamp[i], imo =
      ↪ test_set$imo[i], route_id = test_set$route_id[i],
      ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
      ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
      ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
      ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
  }

  predictors <- c(4,5,7,17,18,22:65)
  for(i in 1:nrow(test_set)){
    pred <- knn.reg(train = train_set[,predictors], test =
      ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
      ↪ = round(sqrt(nrow(train_set))), algorithm="brute")
    predictions_all_noETA <- bind_rows(predictions_all_noETA,
      ↪ data.frame(timestamp = test_set$timestamp[i], imo =
      ↪ test_set$imo[i], route_id = test_set$route_id[i],
      ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
      ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
      ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
      ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
    pred <- knn.reg(train = train_set[,predictors], test =
      ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
      ↪ = round(sqrt(nrow(train_set)))+2, algorithm="brute")
    predictions_allplus2_noETA <-
      ↪ bind_rows(predictions_allplus2_noETA, data.frame(timestamp =
      ↪ test_set$timestamp[i], imo = test_set$imo[i], route_id =
      ↪ test_set$route_id[i],
      ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
      ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
      ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
      ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
    pred <- knn.reg(train = train_set[,predictors], test =
      ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
      ↪ = round(sqrt(nrow(train_set)))-2, algorithm="brute")
    predictions_allmin2_noETA <- bind_rows(predictions_allmin2_noETA,
      ↪ data.frame(timestamp = test_set$timestamp[i], imo =
      ↪ test_set$imo[i], route_id = test_set$route_id[i],
      ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
      ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
      ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
      ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
  }

  predictors <- c(4:7,17,18,22:65)
  for(i in 1:nrow(test_set)){
    pred <- knn.reg(train = train_set[,predictors], test =
      ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
      ↪ = round(sqrt(nrow(train_set))), algorithm="brute")
    predictions_all_course <- bind_rows(predictions_all_noETA,
      ↪ data.frame(timestamp = test_set$timestamp[i], imo =
      ↪ test_set$imo[i], route_id = test_set$route_id[i],
      ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
      ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
      ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
      ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
```

```
      pred <- knn.reg(train = train_set[,predictors], test =
        ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
        ↪ = round(sqrt(nrow(train_set)))+2, algorithm="brute")
      predictions_allplus2_course <-
        ↪ bind_rows(predictions_allplus2_noETA, data.frame(timestamp =
        ↪ test_set$timestamp[i], imo = test_set$imo[i], route_id =
        ↪ test_set$route_id[i],
        ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
        ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
        ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
        ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
      pred <- knn.reg(train = train_set[,predictors], test =
        ↪ test_set[i,predictors], y = as.data.frame(train_set[,19]), k
        ↪ = round(sqrt(nrow(train_set)))-2, algorithm="brute")
      predictions_allmin2_course <-
        ↪ bind_rows(predictions_allmin2_noETA, data.frame(timestamp =
        ↪ test_set$timestamp[i], imo = test_set$imo[i], route_id =
        ↪ test_set$route_id[i],
        ↪ traveltime=round(test_set$traveltime[i]), est_traveltime =
        ↪ test_set$est_traveltime[i], pred_traveltime = pred$pred,
        ↪ pred_ETA=test_set$timestamp[i]+pred$pred*3600, ETA=
        ↪ test_set$eta[i],ATA= test_set$ATA[i], pred_imo=F))
    }

}

errors <- predictions %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_cross_k.eps", width = 15.742708333, height =
  ↪ 14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  ↪ Thesis/Report/Images" )

errors2 <- predictions_all %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    ↪ est_error = abs(mean(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))
```

```
ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_cross_k_all.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors <- predictionsplus2 %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    est_error = mean(abs(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_cross_k+2.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors2 <- predictions_allplus2 %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    est_error = abs(mean(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)), max_error
    = max(abs(traveltime - pred_traveltime)), min_error =
    min(abs(traveltime - pred_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
```

```
    geom_line(aes(y=median_pred, colour="Median_pred_error")) +
    geom_line(aes(y=median_est, colour="Median_est_error")) +
    geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
    geom_line(aes(y=est_error, colour="Avg_est_error"))+
    geom_line(aes(y=max_error, colour="Max_pred_error"))+
    geom_line(aes(y=min_error, colour="Min_pred_error"))+
    scale_colour_manual(name="Line Color",
                        values=c(Avg_pred_error="green",
                          Avg_est_error="lightblue",
                          Median_pred_error="darkgreen",
                          Median_est_error="blue",
                          Max_pred_error="red",
                          Min_pred_error="black")) +
    theme(legend.position="top") +
    labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
      in hours")

ggsave("Results_cross_k+2_all_minmax.eps", width = 15.742708333,
  height = 14.552083333, units = "cm", path =
  "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

errors <- predictionsmin2 %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    est_error = mean(abs(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_cross_k-2.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors2 <- predictions_allmin2 %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    est_error = abs(mean(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
```

```
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_cross_k-2_all.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors <- predictions_noETA %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    est_error = mean(abs(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_noETA_cross_k.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors2 <- predictions_all_noETA %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    est_error = abs(mean(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
```

```
    scale_colour_manual(name="Line Color",
                        values=c(Avg_pred_error="green",
                          ↪ Avg_est_error="lightblue",
                          ↪ Median_pred_error="darkgreen",
                          ↪ Median_est_error="blue")) +
    theme(legend.position="top") +
    labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
      ↪ in hours")

ggsave("Results_noETA_cross_k_all.eps", width = 15.742708333, height
  ↪ = 14.552083333, units = "cm", path =
  ↪ "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

errors <- predictionsplus2_noETA %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_noETA_cross_k+2.eps", width = 15.742708333, height =
  ↪ 14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  ↪ Thesis/Report/Images" )

errors2 <- predictions_allplus2_noETA %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    ↪ est_error = abs(mean(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
```

```
    theme(legend.position="top") +
    labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
      ↪ in hours")

ggsave("Results_noETA_cross_k+2_all.eps", width = 15.742708333,
  ↪ height = 14.552083333, units = "cm", path =
  ↪ "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

errors <- predictionsmin2_noETA %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_noETA_cross_k-2.eps", width = 15.742708333, height =
  ↪ 14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  ↪ Thesis/Report/Images" )

errors2 <- predictions_allmin2_noETA %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    ↪ est_error = abs(mean(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")
```

```
ggsave("Results_noETA_cross_k-2_all.eps", width = 15.742708333,
  ↪ height = 14.552083333, units = "cm", path =
  ↪ "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

errors <- predictions_course %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_course_cross_k.eps", width = 15.742708333, height =
  ↪ 14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  ↪ Thesis/Report/Images" )

errors2 <- predictions_all_course %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    ↪ est_error = abs(mean(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_course_cross_k_all.eps", width = 15.742708333, height
  ↪ = 14.552083333, units = "cm", path =
  ↪ "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )
```

```
errors <- predictionsplus2_course %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_course_cross_k+2.eps", width = 15.742708333, height =
  ↪ 14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  ↪ Thesis/Report/Images" )

errors2 <- predictions_allplus2_course %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    ↪ est_error = abs(mean(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        ↪ Avg_est_error="lightblue",
                        ↪ Median_pred_error="darkgreen",
                        ↪ Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    ↪ in hours")

ggsave("Results_course_cross_k+2_all.eps", width = 15.742708333,
  ↪ height = 14.552083333, units = "cm", path =
  ↪ "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

errors <- predictionsmin2_course %>%
  group_by(route, traveltime) %>%
  summarise(pred_error = mean(abs(traveltime - pred_traveltime)),
    ↪ est_error = mean(abs(traveltime-est_traveltime)),
    ↪ median_pred=median(abs(traveltime - pred_traveltime)),
    ↪ median_est=median(abs(traveltime - est_traveltime)))
```

```
ggplot(errors, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  facet_wrap(~ route,3,2, scales = "free") +
  theme(legend.position="top",strip.text = element_text(size = 5)) +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_course_cross_k-2.eps", width = 15.742708333, height =
  14.552083333, units = "cm", path = "/Users/Ricardo/Dropbox/Master
  Thesis/Report/Images" )

errors2 <- predictions_allmin2_course %>%
  group_by(traveltime = round(traveltime)) %>%
  summarise(pred_error = abs(mean(traveltime - pred_traveltime)),
    est_error = abs(mean(traveltime-est_traveltime)),
    median_pred=median(abs(traveltime - pred_traveltime)),
    median_est=median(abs(traveltime - est_traveltime)))

ggplot(errors2, aes(x=traveltime)) +
  geom_line(aes(y=median_pred, colour="Median_pred_error")) +
  geom_line(aes(y=median_est, colour="Median_est_error")) +
  geom_line(aes(y=pred_error, colour="Avg_pred_error")) +
  geom_line(aes(y=est_error, colour="Avg_est_error"))+
  scale_colour_manual(name="Line Color",
                      values=c(Avg_pred_error="green",
                        Avg_est_error="lightblue",
                        Median_pred_error="darkgreen",
                        Median_est_error="blue")) +
  theme(legend.position="top") +
  labs(x="Hours to arrival in Port of Rotterdam", y="Prediction error
    in hours")

ggsave("Results_course_cross_k-2_all.eps", width = 15.742708333,
  height = 14.552083333, units = "cm", path =
  "/Users/Ricardo/Dropbox/Master Thesis/Report/Images" )

save.image("~/Dropbox/Master Thesis/Data4TU/Results.RData")
```

D

Manual rename file

Table D.1: Manual changes inputted into Excel

IMO	ROUTE ID	WRONG	Right	latitude	longitude
8201648	1	ROTTERDAM NETHERLAND	ROTTERDAM		
8714205	2	BREST FRBES	BREST		
8714205	2	DELWAIDEDOK BEANR	ANTWERP		
8714205	2	NLRTM RTRDM	ROTTERDAM		
9039250	3	TALLINN	MUUGA		
9113745	96	SOUTHSHIELD	SOUTH SHIELDS		
9113745	84	SOUTHSHIELD	SOUTH SHIELDS		
9113745	75	SOUTHSHIELD	SOUTH SHIELDS		
9113733	95	S SHIELDS	SOUTH SHIELDS		
9113745	53	SOUTHSHIELD	SOUTH SHIELDS		
9113745	58	SOUTHSHIELD	SOUTH SHIELDS		
9113745	62	SOUTHSHIELD	SOUTH SHIELDS		
9113745	70	SOUTHSHIELD	SOUTH SHIELDS		
9113745	125	ARHUS	AARHUS		
9121895	80	MONTOIR	MONTOIR DE BRETAGNE		
9129469	67	ELBE	HAMBURG		
9141118	67	TEEPORT	TEESPORT		
9141118	99	REMOVE			
9141118	100	REMOVE			
9141118	125	TEES	TEESPORT		
9141118	131	TEESPORT	ROTTERDAM	51.88007	4.40306900
9141118	141	TEEPORT	TEESPORT		
9162679	29	ECT D	ROTTERDAM		
9162679	35	EUROMAX	ROTTERDAM		
9162679	42	ECT D	ROTTERDAM		
9162679	47	CCT MOERDIJK	MOERDIJK		
9162679	47	ECT D	ROTTERDAM		
9162679	52	ECT D	ROTTERDAM		
9162679	59	APM	ROTTERDAM		
9162679	64	CCT MOERDIJK	MOERDIJK		
9162679	64	ECT D	ROTTERDAM		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9162679	70	ECT D	ROTTERDAM		
9162679	75	MOERDIJK CCT	ROTTERDAM	51.95083	4.072333
9162679	80	DDN	ROTTERDAM		
9162681	70	REMOVE			
9165308	51	ANTWEPEN	ANTWERP		
9169158	2	LE HAVRE FR	ROTTERDAM		
9178537	7	GOTHENBURG	GOTEBORG		
9186388	58	MAASPILOT	ROTTERDAM		
9188506	23	REMOVE			
9188506	26	MAAS PLS	ROTTERDAM		
9188506	26	STEENB PLS	ANTWERP		
9188506	38	ROTTERDAM MAAS PST	ROTTERDAM		
9188506	44	REMOVE			
9188506	45	REMOVE			
9188506	47	REMOVE			
9188506	49	RTM DDE	ROTTERDAM		
9197521	16	REMOVE			
9216834	10	HAMBURGO	HAMBURG		
9216834	12	HAMBURGO	HAMBURG		
9216834	14	ELBE PS HAMBURG	HAMBURG		
9219862	2	NAMBURG	HAMBURG		
9226372		REMOVE			
9227302	10	ANTWERP DRYDOCK	ANTWERP		
9231482		REMOVE			
9231494		REMOVE			
9237371	62	REMOVE			
9237371	89	SENIS	SINES		
9244192		REMOVE			
9244207		REMOVE			
9252096	9	REMOVE			
9256315	14	DUNKERQUE	DUNKIRK		
9256315	21	REMOVE			
9263332		REMOVE			
9264714	65	REMOVE			
9277383		REMOVE			
9286774		REMOVE			
9287699	129	REMOVE			
9287704	27	MONTOIR	MONTOIR DE BRETAGNE		
9287704	28	MONTOIR	MONTOIR DE BRETAGNE		
9287704	29	MONTOIR	MONTOIR DE BRETAGNE		
9287704	30	MONTOIR	MONTOIR DE BRETAGNE		
9287704	31	MONTOIR	MONTOIR DE BRETAGNE		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9287704	34	MONTOIR	MONTOIR DE BRETAGNE		
9287704	36	MONTOIR	MONTOIR DE BRETAGNE		
9287704	37	MONTOIR	MONTOIR DE BRETAGNE		
9287704	38	MONTOIR	MONTOIR DE BRETAGNE		
9287704	40	MONTOIR	MONTOIR DE BRETAGNE		
9287704	42	MONTOIR	MONTOIR DE BRETAGNE		
9287704	43	MONTOIR	MONTOIR DE BRETAGNE		
9287704	44	MONTOIR	MONTOIR DE BRETAGNE		
9287704	45	MONTOIR	MONTOIR DE BRETAGNE		
9287704	46	MONTOIR	MONTOIR DE BRETAGNE		
9287704	50	MONTOIR	MONTOIR DE BRETAGNE		
9287704	51	MONTOIR	MONTOIR DE BRETAGNE		
9287704	52	MONTOIR	MONTOIR DE BRETAGNE		
9287704	54	MONTOIR	MONTOIR DE BRETAGNE		
9287704	55	MONTOIR	MONTOIR DE BRETAGNE		
9287704	56	MONTOIR	MONTOIR DE BRETAGNE		
9287704	57	MONTOIR	MONTOIR DE BRETAGNE		
9287704	59	MONTOIR	MONTOIR DE BRETAGNE		
9287704	61	MONTOIR	MONTOIR DE BRETAGNE		
9287704	67	MONTOIR	MONTOIR DE BRETAGNE		
9287704	67	MONTOJAIRE	MONTOIR DE BRETAGNE		
9287704	68	MONTOIR	MONTOIR DE BRETAGNE		
9287716	36	ARHUS	AARHUS		
9287716	36	KOBENHAVN	COPENHAGEN		
9295397	3	NAPOLIIIIIII	NAPOLI		
9295414		REMOVE			
9297589		REMOVE			
9298612	1	GLUCKSTADT	HAMBURG		
9298612	1	MAAS	ROTTERDAM		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9299501	57	GOTHENBURG	GOTEBORG		
9302243	55	REMOVE			
9302243	61	ROTTERDAM	IMMINGHAM	53.62815	-0.190633300
9302243	65	REMOVE			
9302255	59	REMOVE			
9302255	70	ROTTERDAM	IMMINGHAM	53.62945	-0.19333330
9306079	8	EGH	HAMBURG		
9306079	8	CTA	HAMBURG		
9306079	8	CTB	HAMBURG		
9306079	8	TCT	HAMBURG		
9311842		REMOVE			
9313204	1	TCT	HAMBURG		
9313204	1	CTA	HAMBURG		
9313204	1	CTB	HAMBURG		
9313204	20	GOTHENBURG	GOTEBORG		
9313204	20	GOTHEBURG	GOTEBORG		
9313204	27	CTB	HAMBURG		
9313204	27	ST PETERS	ST PETERSBURG		
9313204	27	WAITING	ST PETERSBURG		
9313204	27	FCT	ST PETERSBURG		
9313204	33	EGH	HAMBURG		
9313204	33	EGH CTB	HAMBURG		
9313204	33	CTT	HAMBURG		
9313204	33	FCT	ST PETERSBURG		
9313204	39	ANTWEREN	ANTWERP		
9313204	39	BUKAI	HAMBURG		
9313204	41	FCT	ST PETERSBURG		
9313204	41	FCT PLP	ST PETERSBURG		
9313204	44	PLP	ST PETERSBURG		
9313204	44	CTSP	ST PETERSBURG		
9313204	47	SFP	ST PETERSBURG		
9313204	47	BUKAI	HAMBURG		
9313204	52	PLP	ST PETERSBURG		
9313204	52	CTA	HAMBURG		
9313216	98	REMOVE			
9313216	168	EGH	HAMBURG		
9313216	168	CTA	HAMBURG		
9313216	168	FCT	ST PETERSBURG		
9313216	168	SFP	ST PETERSBURG		
9313216	171	PLP	ST PETERSBURG		
9313216	171	EGH	HAMBURG		
9313216	171	CTB	HAMBURG		
9313216	171	TCT	HAMBURG		
9313216	171	SFP	ST PETERSBURG		
9313216	171	FCT	ST PETERSBURG		
9313216	175	CTB	HAMBURG		
9313216	175	EGH	HAMBURG		
9313216	175	TCT	HAMBURG		
9313216	175	CTA	HAMBURG		
9313216	175	LAY BY	ST PETERSBURG		
9313216	175	FCT	ST PETERSBURG		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9313216	180	CTB	HAMBURG		
9313216	180	FCT	ST PETERSBURG		
9314973	4	REMOVE			
9314973	6	HUMBURG	HAMBURG		
9315018	66	REMOVE			
9315018	72	REMOVE			
9315018	87	REMOVE			
9315020	1	BRV	BREMERHAVEN		
9315020	13	ROTTERDAM V NOK	ROTTERDAM		
9315020	19	ROTTERDAM V NOK	ROTTERDAM		
9315020	45	REMOVE			
9315032	1	DUNKERQUE	DUNKIRK		
9315032	32	REMOVE			
9315032	39	REMOVE			
9318931	6	ECT	ROTTERDAM		
9318931	35	REMOVE			
9318931	106	REMOVE			
9318931	107	REMOVE			
9318931	110	ROTTERDAM CCT M	MOERDIJK		
9318931	110	ROTTERDAM ECT D	ROTTERDAM		
9318931	120	KRAMER	ROTTERDAM		
9319571	1	GOTHENBURG	GOTEBORG		
9319571	13	BREVIC	BREVIK		
9319571	13	GOTHENBURG	GOTEBORG		
9319571	24	GOTHENBURG	GOTEBORG		
9319571	33	GOTHENBURG	GOTEBORG		
9319868	62	REMOVE			
9322566	150	REMOVE			
9323467	11	REMOVE			
9323467	20	TALLINN	MUUGA		
9328027	9	REMOVE			
9328637	9	REMOVE			
9328637	13	GDINYA	GDYNIA		
9328637	26	TALLINN	MUUGA		
9333357	90	GOTHENBURG	GOTEBORG		
9333357	197	TEES	TEESPORT		
9333357	201	TEES	TEESPORT		
9333357	208	TEES	TEESPORT		
9333357	230	REMOVE			
9333369	12	DUNKERQUE	DUNKIRK		
9333369	21	REMOVE			
9333383	34	REMOVE			
9339026	243	REMOVE			
9339038	160	GOTHENBURG	GOTEBORG		
9341964	51	TALLINN	MUUGA		
9341964	57	TALLINN	MUUGA		
9341964	110	GOTHENBURG	GOTEBORG		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9341964	174	GOTHENBURG	GOTEBORG		
9341976	31	DUNKERQUE	DUNKIRK		
9344253	78	MC BUOY	ROTTERDAM		
9344710	16	DUNKERQUE	DUNKIRK		
9344710	19	ANTWERP STEENBANK	ANTWERP		
9344710	19	DUNKERQUE	DUNKIRK		
9344710	20	DUNKERQUE	DUNKIRK		
9344722	3	REMOVE			
9345996		REMOVE			
9349215	45	REMOVE			
9349215	59	REMOVE			
9349215	73	REMOVE			
9349215	85	REMOVE			
9349215	91	REMOVE			
9349215	94	REMOVE			
9349215	106	REMOVE			
9349227	51	REMOVE			
9349227	65	REMOVE			
9349227	73	REMOVE			
9349227	83	REMOVE			
9349825	1	ANTWERP DRY DOCK	ANTWERP		
9349825	2	ANTWERP STEENBANK	HAMBURG		
9351593		REMOVE			
9354351	1	REMOVE			
9354351	36	IE DUBLIN	DUBLIN		
9354351	49	REMOVE			
9354351	52	REMOVE			
9354351	54	REMOVE			
9354478	59	REMOVE			
9354351	63	REMOVE			
9354351	67	REMOVE			
9354478	19	DEBHV	BREMERHAVEN		
9354533	25	REMOVE			
9355446	78	RTM EEMHAVEN	ROTTERDAM		
9355446	208	TEES	TEESPORT		
9355446	214	TEES	TEESPORT		
9355446	281	SOUTHSHIELD	SOUTH SHIELDS		
9355460	338	RDAM	ROTTERDAM		
9355812	11	RUPLP	ST PETERSBURG		
9360972	26	REMOVE			
9360972	57	CCT	BREMERHAVEN		
9360972	57	KRAMER	ROTTERDAM		
9360972	93	MONTOIR	MONTOIR DE BRETAGNE		
9360972	93	DUNKERQUE	DUNKIRK		
9360996	1	REMOVE			
9365984	30	REMOVE			
9369007	170	TEES	TEESPORT		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9369007	204	REMOVE			
9369007	216	GRANMOUTH	GRANGEMOUTH		
9376024	13	ST PETERSB	ST PETERSBURG		
9376024	13	VUOSAARI	HELSINKI		
9376024	29	REMOVE			
9395551	54	S SHILDS	SOUTH SHIELDS		
9395551	83	RST	ROTTERDAM		
9395551	92	ANTWERP STEENBANK	ANTWERP		
9395551	99	ANTWERP STEENBANK	ANTWERP		
9395575	106	TYNE	SOUTH SHIELDS		
9395575	160	ANR	ANTWERP		
9395575	164	ANR	ANTWERP		
9395575	179	ANR	ANTWERP		
9395575	181	ANR	ANTWERP		
9395575	251	SOUTHSHIELD	SOUTH SHIELDS		
9395575	266	DUB	DUBLIN		
9396696	9	REMOVE			
9399753	1	REMOVE			
9404089	254	REMOVE			
9404089	266	TIL	TILBURY		
9404089	267	TIL	TILBURY		
9404089	268	TIL	TILBURY		
9409209	4	REMOVE			
9410765	6	DUNKERQUE	DUNKIRK		
9418652	7	IZMIR	IZMIR (SMYRNA)		
9428205	88	REMOVE			
9429194	144	RTM	ROTTERDAM		
9429211	10	ULU	UST-LUGA		
9429211	10	SFP	ST PETERSBURG		
9430193	1	REMOVE			
9437191	18	DUNKERQUE	DUNKIRK		
9440576	93	REMOVE			
9440590	46	REMOVE			
9440605	164	TALLINN	MUUGA		
9440605	291	GOTHENBURG	GOTEBORG		
9440605	308	GOTHENBURG	GOTEBORG		
9440605	331	GOTHENBURG	GOTEBORG		
9440605	356	GOTHENBURG	GOTEBORG		
9445904	3	REMOVE			
9448695	123	REMOVE			
9454448	9	DUNKERQUE	DUNKIRK		
9467275	8	STEENBANK PBG	ANTWERP		
9467299	10	ANR	ANTWERP		
9483358	82	LENINGRAD	ST PETERSBURG		
9483358	123	LENINGRAD	ST PETERSBURG		
9483358	123	SPB NMT	ST PETERSBURG		
9483358	127	NOK	KOTKA		
9483358	127	TALLINN	MUUGA		
9483358	135	TALLINN	MUUGA		

Continued on next page

Table D.1 – continued from previous page

IMO	ROUTE ID	Wrong	Right	latitude	longitude
9483358	153	NOK	ROTTERDAM		
9483358	156	REMOVE			
9483695	73	CTA	HAMBURG		
9483695	82		ZEEBRUGGE		
9483695	98	TALLINN	MUUGA		
9501332	3	GB SOUTHAMPTON	SOUTHAMPTON		
9502960	6	ROTTERDAM NETHERLAND	ROTTERDAM		
9504035	129	REMOVE			
9504035	164	RST ZZ	ROTTERDAM		
9525912	9	FXT	FELIXSTOWE		
9584865	81	REMOVE			
9584865	120	REMOVE			
9584865	134	ARHUS	AARHUS		
9584865	141	ARHUS	AARHUS		
9584865	141	GOTHENBURG	GOTEBORG		
9584865	154	REMOVE			
9605267	5	REMOVE			
9606326	5	DEWVH	WILHELMSHAVEN		
9613020	4	REMOVE			
9619426	9	REMOVE			
9619933	8	GOTHENBURG	GOTEBORG		
9619971	6	GOTHENBURG	GOTEBORG		
9632064	2	GOTHENBURG	GOTEBORG		
9632064	4	GOTHENBURG	GOTEBORG		
9637258	6	ROTTERDAM NEDERLAND	ROTTERDAM		
9637260	6	ELBE P	HAMBURG		
9665619	9	NLRDM	ROTTERDAM		
9665619	11	GBFLX	FELIXSTOWE		
9665619	13	GBFLX	FELIXSTOWE		
9665633	14	GBFLX	FELIXSTOWE		
9665633	14	NLRDM	ROTTERDAM		
9665633	16	GBFLX	FELIXSTOWE		
9665645	13	ROTTERDAM MAAS PLT	ROTTERDAM		
9667150	14	REMOVE			
9667186	11	REMOVE			
9683477	36	REMOVE			
9684653	2	NLRTM MAASS	ROTTERDAM		
9684677	1	PIRAEUS GREECE	PIRAEUS		
9695133	4	NLRTL	ROTTERDAM		
9708784	2	BE ZEEBRUGGE WANDELA	ZEEBRUGGE		
9708784	2	BE ZEEBRUGGE WANDEL	ZEEBRUGGE		
9708784	2	BE ZEEBRUGGE	ZEEBRUGGE		