VRIJE UNIVERSITEIT AMSTERDAM

MASTER THESIS

BUSINESS ANALYTICS

Classifying vessel types based on AIS data

Author:
L. Westerdijk

External supervisor:
M. Beekveld

Graduation supervisor:
Prof. Dr. A.E. Eiben

Second reader:
Dr. E.N. Belitser

28th February, 2019

Classifying vessel types based on AIS data

L. Westerdijk

Graduation thesis
Internship report

Vrije Universiteit Amsterdam
Faculty of Science
Business Analytics
De Boelelaan 1105
1081 HV Amsterdam

Thales Hengelo
Zuidelijke Havenweg 40
7554 RR Hengelo

February 2019

Preface

This Master Project serves as the graduation project, which is a compulsory part of the Masters degree program Business Analytics at the Vrije Universiteit Amsterdam. The graduation thesis has been produced as part of the final product of the mandatory six-month internship. This internship and research have been executed at an external organisation. The internship assignment should be appropriate to the Business Analytics program, which means it should be in line with the courses and should be of scientific value to the university. Moreover, the final product should have a practical value for the external organisation. The overall purpose of the master thesis is to enable the student to develop deeper knowledge, understanding and capabilities as regards Business Analytics based on the independent work.

This research is conducted within the external host organisation Thales Hengelo. Thales operates worldwide in the field of aerospace, space, transportation, defence and security. The location in Hengelo is primarily involved in defence and security - this project is implemented in their naval domain. The context of this research falls within the scope of the Above Water Warfare section and the CMS applications department.

The terms of reference include implementing machine learning classification techniques that can predict the type of vessel. Based on incoming AIS data the final model should be able to classify the vessels nearby. This prediction is considered an advice for the human operator on board of the naval vessel, which should help reduce the operators workload.

In truth, I am pleased with the final result of this graduation thesis. I have achieved this level of success with the support of my supervisors. First of all, I would like to thank Marcel Beekveld, my supervisor of Thales, for guiding and supporting me during this internship. Especially the intense feedback sessions in the final stages of writing this thesis have been helpful. Moreover, I would like to thank Jan Egbert Hamming, Thomas ter Wijlen, Boudewijn Geerink and Franck Maarse for their involvement during my research. I would also like to thank Guszti Eiben and Eduard Belitser for, respectively, being my first supervisor and my second reader of this thesis. Finally, a special thanks goes to my family for their loving support during my studies.

Larissa Westerdijk
February 2019

Management summary

Classifying vessel types based on AIS data

by L. Westerdijk

The goal of this research is to support Thales in deciding whether machine learning can contribute to classifying vessel types. The classification of the vessel indicates labelling the vessel with the right category, e.g. a cargo, a tanker or a fishing vessel. Currently, several Navies use the Combat Management System developed by Thales in order to manually classify the vessels in the surrounding area. During missions the human operator on board of the naval vessel carries out this task to provide situational awareness by analysing incoming information of each vessel. However, this takes a significant amount of time and effort. Therefore, it is desirable to develop a product that can support the operator in making decisions, preventing to miss important events, reducing the workload and keeping focus. Since Thales is not in the possession of worldwide data, a global product cannot be realised. Consequently, a product can only be developed if the underlying statistical models can be trained to accurately represent local conditions and mission types. Machine learning techniques seem to fit this problem statement, which leads to examining the following research question.

Is it possible to develop a sufficiently reliable product that classifies vessels using machine learning?

The above-mentioned research question will be answered by researching which machine learning techniques work best and if the results are good enough. First, the data is collected from original AIS messages, which are messages that vessels transmit, containing information about the vessel. The data has been analysed in order to examine how it should be used to achieve the highest predictive power of the implemented models. Since the quality of the data is the most important aspect towards a sufficient classification model, the dataset is enlarged by extra engineered features that possibly contribute to the predictive power. Three conceptually different but suitable machine learning techniques have been considered in order to investigate which kind of algorithm fits this problem best. These techniques are the random forest, the support vector machine and the artificial neural network. The performance of each technique is optimised by implementing various combinations of selected explanatory variables, chosen parameter values and compositions of the target variable. Each final model performance is evaluated using the accuracy and F1 score, with the primary aim to optimise the accuracy.

The results have shown that, within the restrictions of this research, the trained support vector machine achieves the best performance. This is based on both the accuracy originating from the test set and the confidence interval. Moreover, both the support vector machine model and the random forest model perform significantly better than the artificial neural network model. However, the absolute accuracy value along with the confidence interval of the support vector machine model are not considered sufficiently reliable for developing a product that classifies vessels. First of all, the accuracy given the data is too low, since it does not support the operator enough and rather causes confusion. Secondly, the robustness of the model performance, derived from the corresponding confidence interval is insufficient, because the interval range is relatively wide.

It is concluded that the support vector machine model does not seem to produce usable predictions because of the quality of this specific dataset. The dataset covers a relatively short period of time and a too specific geographical location, which provides insufficient ground truth for each vessel type category. Consequently, a key recommendation for improving the predictive power of a vessel classification model is to make sure the dataset contains enough evidence for each vessel type category. Once a sufficiently reliable product can be realised to classify vessels, the prediction should always be considered as an advice to the operator rather than replacing the operators decision.

Contents

1 Introduction

Thales Group is a worldwide company active in the field of aerospace, space, defence, security and transportation. 'Thales helps its customers create a safer world by giving them the tools they need to perform critical tasks. World-class technology, the combined expertise of 65,000 employees and operations in more than 50 countries have made Thales a key player in keeping the public safe and secure, guarding vital infrastructure and protecting the national security interests of countries around the globe.' [30]

Thales Netherlands is represented by the locations in Hengelo, Huizen, Delft and Eindhoven. Employees at Thales Huizen contribute to communication systems, communication networks, solutions for cyber security, and installation and maintenance for the national OV-chip equipment. The Research & Development department is located at Thales Delft. The focus is on R&D activities in the field of radar technology and radar systems, and close collaboration is realised with TU Delft. Thales Eindhoven represents Thales Cryogenics and focuses on developing and fabricating cryogenic refrigeration systems. Finally, the headquarter Thales Hengelo is primarily involved in safety and security, both in the civil and defence realm. One of its main markets is the worldwide naval defence and maritime field. The location in Hengelo is world leader in the latest and most innovative technologies like radar systems for naval vessels [28]. The section Above Water Warfare at Thales Hengelo is developing a Combat Management System (CMS) called TACTICOS. Twenty-four navies worldwide use this system on board of naval vessels tailored to maritime security and defence tasks. TACTICOS consists of multiple Multi-function Operator Consoles displaying high-resolution information received from sensors. An example of such a console is illustrated in Figure 10.1 in the Appendix. A clear picture showing all technical specifications around TACTICOS is attached to the Appendix as well, shown in Figure 10.2. 'It is the central command and decision-making element of a naval vessel combat system. Its function and performance are critical to the operational effectiveness of a naval vessel.' As is stated by the TACTICOS brochure [25].

A specific purpose of the CMS is to create situational awareness of the surrounding area. Identifying and displaying as much information of the nearby vessels provides the basis for the operator to create this situational awareness. During missions, one of the human operators tasks on board of the naval vessel is to provide the situational awareness by identifying and classifying the vessels in the nearby area. For this, the operator analyses each vessel based on the incoming information that is displayed on the console. One data source originates from AIS messages, which are messages containing static and dynamic information about the vessel transmitted by the vessel itself. The CMS checks the dynamic kinematic data from the AIS messages with similar data originating from the sensors. Due to this fusion with data from on board sensors, the kinematics of the AIS track update can be assumed to be correct. The classification of the vessel indicates labelling the vessel with the right category, e.g. a cargo, a tanker or a fishing vessel. The identification defines the allegiance of the vessel, meaning it will be identified as either friendly, hostile, neutral, ads friendly, suspect or unknown. First, the classification of the vessel type is determined and subsequently the identification of the vessel. This task is time-consuming, requires a lot of domain knowledge and particularly classifying a vessel can be difficult in some cases.

Therefore, the ASCI research group at Thales has developed a Bayesian network model that can support the operator in making decisions, preventing to miss important events, reducing the workload and keeping focus. The model automatically identifies and classifies nearby vessels, based on the information provided by the TACTICOS system. Because of certain deficiencies concerning this model, as will be further explained in Section 3.1, Thales had the ambition to develop an alternative model. The Dutch navy has indicated that a human operator easily conducts the identification of a vessel being friendly, hostile, neutral, suspect or unknown. However, classifying the vessel type requires more valuable time and effort. Machine learning is an approach that can provide the TACTICOS system the ability to learn certain prediction tasks from experience (i.e. training data containing the explanatory variables and the target variable) without being explicitly programmed. Because of the availability of labelled data, this raises the question of the usability of machine learning for this specific problem. Machine learning can be deployed for creating a predictive model that classifies the target variable vessel type. This predictive model is, preferably, integrated into the TACTICOS system, since this makes the product more attractive

for worldwide navies.

Hence, the business aim for Thales is to develop a product that is able to classify vessel types, based on data that is available in the TACTICOS environment. In order to investigate whether such a product is beneficial, this research focuses on implementing a number of machine learning classification algorithms for the naval domain to predict vessel types. This is formulated in the following research question.

Is it possible to develop a sufficiently reliable product that classifies vessels using machine learning?

A sufficiently reliable product is a rather vague concept, which depends on specific needs of Thales. Here sufficient indicates whether the final product is mature enough for bringing it into the market. More precisely, it primarily means that the predictive power and robustness of the final model is high enough. In addition, the product should be suitable to be implemented in the organisation. Both theoretical and technical aspects should be judged in consultation with Thales employees with appropriate domain knowledge. A clear answer to this question involves examining several sub questions, which will be divided in the three categories: data related, algorithm related and organisation related. These will be specified in Section 3.2 Research questions. In case the final model is perceived as a potentially appropriate product, the predictions of the final classification model are, then, considered as an advice for the human operator on board. Hence, the model will contribute to providing situational awareness by automated vessel classification. It should be noted that the classification model only supports the operators decision and does not replace the operator. Once the model is suitable for implementation on board of a marine vessel, the optimal application requires the data to go through a filter to separate abnormal vessels from the dataset. These suspicious and irregular behaving vessels need the attention and analysis of the operator regardless and, therefore, do not need to be classified by the model.

The remainder of this thesis is divided into nine chapters. First, the target audience of this thesis will be addressed. Section 3 describes the final graduation assignment completed at Thales Hengelo. The following section discusses the relevant literature in the maritime domain. The data description, analysis and pre-processing will be treated in Section 4. Subsequently, Section 6 contains the explanation of the underlying theoretical methods, the evaluation and the experimental setup. The results of the experiments will be provided and evaluated in Section 7, followed by the discussion in Section 8. Finally, the thesis will end with the conclusions and recommendations.

Most sections of this thesis are essential for understanding the research as a whole. However, some sections can be omitted if time is limited. In that case the advice is to omit Section 2 Target audience, Section 4 Relevant literature, Section 6.1 Clustering method and Section 6.2 Predictive classification methods. The latter two sections are for gaining in-depth understanding of the implemented machine learning techniques.

2 Target audience

Since this master thesis and the associated research are implemented within an external organisation, the target audience consists of multiple stakeholders. The main focus is to transfer the acquired knowledge to the external organisation Thales. The stakeholders at Thales can be divided into three different groups, with each of them having their own reason to be interested. Apart from the engineers, researchers and managers at Thales, the Vrije Universiteit Amsterdam is also one of the stakeholders of this thesis.

The general interest of Thales as a stakeholder is to acquire knowledge about the implementation of machine learning. Thales Hengelo is in the initial phase of using machine learning within its business processes. A lot of theoretical knowledge is available at the Research & Development department at Thales Delft. However, they lack knowledge about the actual implementation within this context. The researchers would, therefore, benefit from this thesis by learning about machine learning within this specific domain. Conversely, the theoretical knowledge in the operational business side located in Hengelo is not sufficient as it is in the location in Delft. Therefore, as stakeholders the engineers would benefit from this thesis by learning about machine learning in general. This enables this thesis to act as a bridge between the theoretical and operational sides of Thales Netherlands, necessary to gain expertise in this field. The third stakeholder group at Thales are the managers, and more specifically, the innovation managers and product managers. A manager is required to know about the activities in the concerning department. This thesis could be construed as an introduction to machine learning and makes a comprehensive statement regarding the possibilities of machine learning in the two departments.

The remaining stakeholder is the university Vrije Universiteit Amsterdam, represented by the first supervisor and the second reader. Since they have extensive theoretical knowledge and experience, their special interest is in the implementation of this specific problem. Moreover, the scientific level must be high along with a clear, representative research outline.

In order to make the content of this thesis sufficiently interesting and essential for all stakeholders, both context and theory are dealt with in detail. This is, because Thales and the university have rather different priorities. The main interest of both is in the theoretical implementation in this specific context. Yet Thales focuses more on the theory, while the university focuses on the context.

3 Research description

This section outlines all aspects of the research description, starting with the motivation for this research. The initial idea for this research originated from previous work at Thales. First, this previous work will be clarified. Based on those findings the main research question and related sub questions of this study will be established. For this research to be successful, the research question should have a clear and unambiguous answer. The actual answer to this question depends on the extent it will meet the success criteria. The scope of the research is clearly stated in order to correctly elaborate on the approach. Finally, this section highlights the requirements and tool selection.

3.1 Existing internal work

The ASCI innovation team at Thales has developed the Automated Surface Classification and Identification (ASCI) system. Results of ASCI are part of the CMS TACTICOS and determine the classification and identification of the vessels nearby, based on incoming data from sensors. The previous underlying model was based on a model from 1998, which is identified with several serious deficiencies that make the model complex and hard to maintain. This is due to the need for a priori probability distributions. Moreover, this model was built during the cold war, which makes the purpose for using the model not comparable with the purpose nowadays. Back in those days, the composition of vessels and the focus during the missions were different. The main focus for the human operators during missions nowadays is related to detection of illegal fishing, drug smuggling or refugee ships, while the focus used to be combat related like detection of dangerous war ships. This makes the model inaccurate. Considering these factors, the old model is impractical to adapt to the required small updates and current situations. Hence, the need emerged for an alternative model that is able to cope with new requirements.

The internal work has led to the bachelor assignment of a previous student intern at Thales. During this internship the focus was on developing a Bayesian network that is able to predict the identity of the vessels. The classification of the vessels is excluded from this research. A Bayesian network satisfies all new requirements, lacking in the old model. The Bayesian network algorithm is trained using few features that can be extracted from the concerning AIS data. AIS data will be explained in more detail in Section 5.1. A typical characteristic of naval data is scarcity, which means expectation-maximisation is best suited to do the parameter learning for creating the final model. Expectation-maximisation is an algorithm that estimates parameter values based on the maximum-likelihood estimates, which makes it an appropriate technique to use for naval data. This final Bayesian network model has the input nodes cruising speed, special area and vessel type with as target node the identity that can take one of the two categories non-aligned or unknown. Despite the limited amount of data, the results of predicting the identity, i.e. the vessels allegiance, show that the discriminating capability between the vessel identities is high. The predictions for this specific situation achieve an accuracy of approximately 99.64%. However, the research is restricted in its implementation. First of all, the identity of the vessels is limited to two categories. In addition, the input variables are categorical due to binning the continuous variables and the input variables are few in terms of quantity. This bachelor assignment only considered a Bayesian network algorithm and, finally, the focus was on identifying the vessels identity. This raises the question as to what the consequences would be in case more input variables are used, in case continuous variables are used in combination with other machine learning techniques and in case the target is to classify the vessel type. Especially this last issue is intriguing, since the Dutch Navy has emphasised that identifying the vessels is an easy job. Yet there is need for a model that is able to support the operator with classifying the vessels.

Therefore, the main objective of this research is to investigate whether machine learning techniques are valuable for creating a product that is able to classify the vessel type.

3.2 Research questions

In this research, several machine learning techniques are explored and applied within the naval domain in order to answer the following research question.

Is it possible to develop a sufficiently reliable product that classifies vessels using machine learning?

This research question can be answered with the help of the following sub questions, which are categorised as being data related, algorithm related or organisation related.

Data related

- *Is the quality of the data sufficient?*

Algorithm related

- *What validation criteria apply best for this specific problem?*
- *Given the available data, which machine learning techniques apply best to classify vessels?*
- *How robust are the performance values of the models obtained by the machine learning techniques, with regard to the parameter values and the geographical dependence?*

Organisation related

- *What skills and knowledge should be present in the organisation to develop the desired product?*

3.3 Success criteria

To ensure the success of this research a well-founded answer to the research question is essential. Since the research question is dependent on and supported by the above-mentioned sub questions these need clear, preferably positive, answers as well. In order to achieve clear answers, it is necessary to be able to explain why these answers apply. The success criteria can be derived from the sub questions and are explicitly mentioned and clarified in the next four paragraphs.

The most important factor that determines the ability to develop a sufficiently reliable product is the predictive power of the obtained models. This predictive power can be evaluated by several performance measures. The statistical evaluation criteria that are considered during this research will be explained in detail in Section 6.3. The quality of the data has the most influence on the model its predictive power. In case the dataset does not consist of distinct patterns, the discriminating capability between the vessel types is low which results in a low predictive power. It is, therefore, important to be certain about the data quality. Data quality can be tested a priori and a posteriori. A priori data quality is determined by pure statistical analysis of the data, which will be discussed in Section 5.2 Data analysis. A posteriori data quality, however, is determined by the discriminating capabilities of the considered models. Their actual performance is evaluated in Section 7 Experimental results and evaluation.

Another success criterion involves the adaptability of the organisation and, therefore, whether the desired product can be implemented and included at Thales. Whether the product would fit in the organisation can be evaluated after determining what skills and knowledge is needed to develop, maintain and support the product. If those requirements are reasonable and useful enough for Thales to arrange, the product can be implemented in the organisation.

Besides the importance of a high predictive power of the model performance, the robustness of the model is a major factor for developing a product as well. The robustness can be evaluated based on two different adjustment approaches. First, the robustness of the model will be evaluated based on changing the parameter values and observing the effect on performance measures. This parameter robustness analysis will be explained in more detail in Section 6.4.2 Parameter robustness analysis and evaluated in Section 7 Experimental results and evaluation. The other approach involves the geographical robustness of the model. The geographical dependence can be tested using geographical different datasets but equivalent features and model parameter values. The exact approach will be described in Section 6.4.4 Geographical dependence analysis.

The final criterion that affects the sufficiency of the desired product is the extent to which outliers can be detected and separated. Determining whether an instance is an outlier or not is a difficult task because it is hard to know what threshold can be applied best. The filter that should be implemented will cover the biggest part of the task to detect abnormal vessels, which is outside the scope of this research. The specific implementation of detecting distinctive data instances will be discussed in Section 5.3.5 Outliers.

3.4 Scope

This research is defined within a certain scope. The request originating from the business involves the aim to develop a model that classifies the ordinary vessels. Specific data instances that indicate abnormal vessel behaviour or characteristics are removed (by the filter) from the dataset and left for the human operator to analyse. AIS data will be used to create the classification models, since this data is labelled and usable for supervised learning techniques. Thales is in the possession of data obtained from the sensors of two Dutch marine vessels. However, sensor data does not contain the specific vessel type information that is crucial to this problem. Sensor data along with the AIS data can be used for applying semi-supervised learning, but is outside the scope of this research. The initial task of validating the target labels within this research utilises an unsupervised learning technique, since the dataset is rather large to manually label and validate the vessel types. This will be limited to applying the k-means algorithm, because k-means is one of the most popular clustering algorithms. The supervised learning algorithms that are needed for this research are limited to the implementation of the three techniques random forest, support vector machine and neural network. These three algorithms are common and significantly different techniques that are suited for a classification problem.

3.5 Approach

The approach of this research is divided in two parts, a preliminary investigation and the main study. Each of the subsequent sections adds a part of the whole research process. However, this research starts with a preliminary investigation that serves as introduction into the naval domain and the working environment and to ensure the main assignment. This investigation is shortly explained below.

This initial investigation is based on the previously developed Bayesian network and aims to examine whether an interpolated dataset performs significantly different compared to the original dataset with the same size. The reason for this small investigation originates from the fact that naval data is sparse because of confidentiality and short missions. Some machine learning techniques, especially clustering algorithms, need a vast amount of data to be able to have a good performance. Therefore, expanding a small dataset in order to make it useful for specific machine learning algorithms is rather convenient at the business domain of Thales. The effect on the model performance can tell whether this adaption of a dataset is appropriate and useful. In this little lab experiment the categorical variables between each two instances of each unique vessel are interpolated. The result is the augmented dataset. This augmented dataset and the original dataset, both of the same size, are put in the developed Bayesian network, which results in the corresponding performance. The conclusion of the results can be extended to the main study. The results show no significant difference between the performance measures, which means using an interpolated dataset has no negative consequences in this specific situation. However, the results specifically apply to a Bayesian network model rather than models resulting from other machine learning algorithms. Moreover, the exact data differs from the data that will be used during the main assignment. The few features used in the Bayesian network model are categorical, which means interpolating the instances results in nearly always, if not always, the exact same interpolated instance. Hence, the dataset is just expanded with equal rows.

The approach of the main study involves all necessary steps in a machine learning chain. Once the data is collected, it is analysed in order to interpret the data. The analysis is important for possible future process decisions in the interest of optimising the final model performance. For example, the analysis shows what is needed during the data pre-processing. The data needs cleaning before it will be used to train the machine learning techniques, because the data should be as close to reality as possible. Cleaning the data means removing duplicates, handling missing values, removing outliers and creating ground truth labels. The remaining steps of data pre-processing are feature engineering, possible feature transformation and feature selection. All data pre-processing steps will be described in more detail in Section 5.3. All features except the target feature *vessel type* serve as possible input variables of the training process, the target feature is the unobserved variable that needs to be determined. The final model will be achieved through an iterative optimisation process. Within this process the parameters will be tuned, input features will be tuned and possible other operations will be applied that are necessary for optimising the performance. The exact details of this optimisation process will be discussed in Section 6.4.1. Each model performance needs an evaluation, involving the use of several measurement methods. The evaluation of each model performance will be treated in Section 7. All previously mentioned steps that are considered

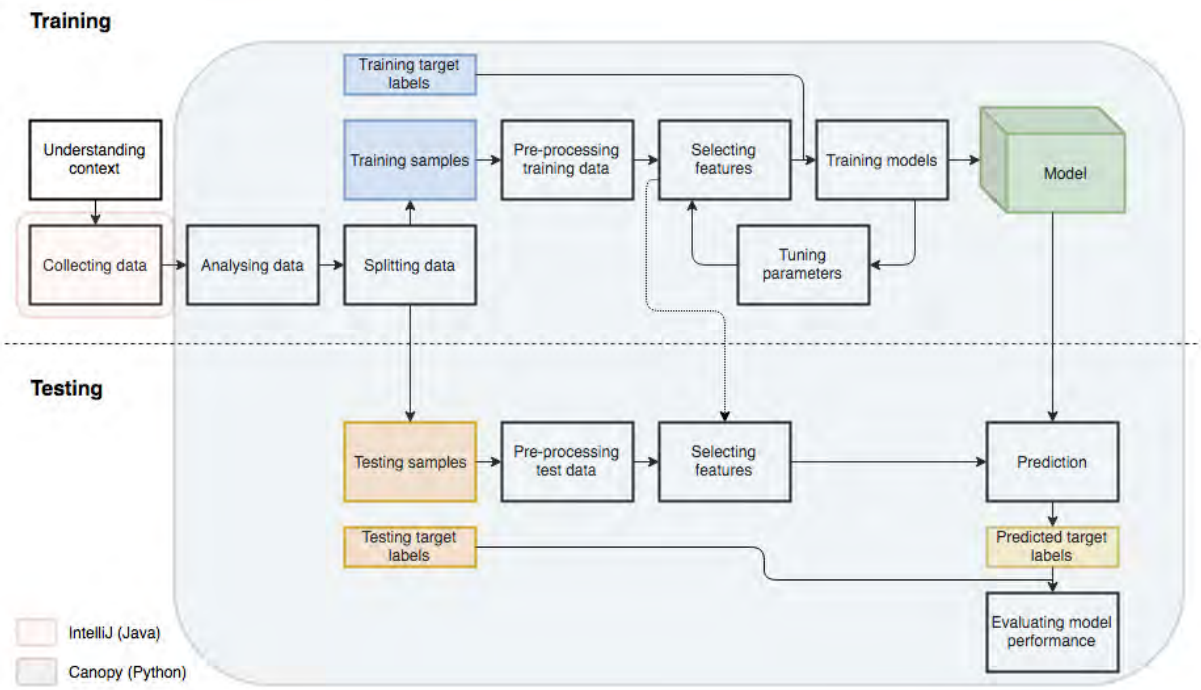

FIGURE 3.1: Structure of machine learning implementation

during the main study can be visualised as in Figure 3.1, which shows the data flow of the whole process.

3.6 Requirements and tool selection

In order to make this research reproducible and implementable and motivate the tooling considerations, this subsection presents the tooling selection. Thales offered free choice of tooling and programming language, meaning there are no predetermined requirements. However, for the purpose of good guidance and usage of the results within the organisation, programming languages that are known and applied at Thales are preferred. Moreover, two other aspects are important considering the selection of tools. First of all, the choices are committed to the technical environment that Thales has available. All the programming code should be executed under a secure CentOS release 6.10 (Red Hat Linux) environment, which entails many dependencies. Those dependencies are the reason why the considered alternative, the free software environment R, cannot be used for this research. Programming code could technically also be executed in the Windows environment, but the confidential data is not allowed to be used in this insecure environment. Secondly, the maturity of the tools and packages is essential. This involves the stability, the size of the community and the level of documentation. Considering both aspects, the processes within this research, starting with the data analysis, will be implemented within Canopy. This platform provides several appropriate packages with Python as the programming language. The collection of data is executed in IntelliJ, using and modifying the existing Java code. The concerning output is the input for the data analysis and remaining machine learning steps within the Canopy environment. The preliminary investigation builds on the existing Bayesian network model of the previous intern. Therefore, the same environment of Java within IntelliJ and NeticaJ is used. The list of established requirements about the tooling are shown in Table 3.2. An overview concerning the selected tools is shown in Table 3.1.

TABLE 3.1: Tool selection

Assignment type	Task	Environment	Packages
Preliminary investigation	All steps	IntelliJ (and NeticaJ)	EsiAisParser, Netica
Main research	Collecting data	IntelliJ	EsiAisParser
	Remaining ML steps	Canopy	Scikit-learn, Pandas, Numpy, SciPy, Matplotlib, Seaborn, Shapely

TABLE 3.2: List of requirements

Number	Requirement
1	Tool should fit in Thales environment
2	Tool should offer a great range of machine learning algorithms
3	Tool should offer metrics to evaluate trained models
4	Tool should offer mathematical calculations
5	Tool should offer statistical tests
6	Tool should offer adequate visualisation possibilities

4 Relevant literature

This section briefly discusses some relevant literature that is available in the naval domain. Machine learning is a much-studied topic in this context and is particularly focused on using AIS data. Especially the use of AIS data is popular in those researches, since it contains relevant information about the behaviour of the vessels.

Many studies in the maritime domain focus on anomaly detection. The studies Mascaro et al. (2010) [20], Cheng et al. (2017) [24], Laxhammar (2008) [18] and Pallotta et al. (2013) [22] all did research on the detection of anomalies in the vessel behaviour, using different approaches. Mascaro et al. used Bayesian networks to detect anomalous vessels. The two developed models both use a different approach; the time series approach works at the message level while the track summary approach includes all messages in the track. Apart from AIS data, the research of Mascaro et al. also uses extra data about the weather, about the day and about the ship. The study of Cheng et al. aims to identify smuggling techniques based on anomaly detection methods. The acquired knowledge about the modelled behaviours of interest is used to identify smugglers. A one-class support vector machine is used to label unseen data as either anomalous or not and detect possible smuggling vessels. The feature engineering process of this research takes particular account into the behaviour of the vessels with the aim to reveal potential anomalies. Laxhammar has developed an anomalous detection system for sea surveillance using a Gaussian Mixture Model and greedy Expectation-Maximisation. Finally, Pallotta et al. proposes the methodology Traffic Route Extraction and Anomaly Detection (TREAD) to understand traffic patterns to create maritime situational awareness. The approach converts raw data into information that supports decisions. One application of this method is to detect anomalies based on a statistical model, which uses single trajectory points and involves time information to include information of successive data points.

Apart from the anomaly detection, the TREAD methodology of Pallotta et al. (2013) also addressed the application of route predictions of the vessels. Another research that focuses on predicting vessel trajectories is the research of Young (2017) [31]. The two different statistical learning algorithms random forest and neural network are used to predict future vessel locations based on the historic data originating from AIS. The previous mentioned researches are particularly useful to get an impression of the current state of machine learning in the naval domain using AIS data. Moreover, the researches are useful for creating ideas about engineering adequate features that include information about the vessels behaviour.

However, these researches do not have the aim to classify the vessel type, which is the aim of this master thesis. Falcon et al. (2014) [8] does attack this problem using fuzzy-rough decision trees to learn about the behaviour of vessel types. Although the aim of Falcon et al. (2014) is similar to the aim of this research, the approach and corresponding methods are different. The AIS messages of each vessel are summarised into a summary vector in the form of descriptive track features, which contain kinematic, static and environmental information. Based on the summary vector the trained model correctly predicts the vessel type in over 80% of the cases. The exact approach is not appropriate for the real time implementation Thales desires, since predictions should be made during the receiving process of the incoming messages rather than once all messages are received.

The approach of Ljunggren (2017) [19] classifies vessel types for the more specified aim to find erroneous vessel types in the AIS data. Ljunggren considers two different pre-processed datasets, both originating from roughly 1100 hours of recorded AIS messages. Only vessel trajectories consisting of more than 30 AIS messages are considered. Moreover, all trajectories are cut into equally long trajectory samples. The exact data pre-processing steps also remove vessels that have too low median speed and remove certain vessel types due to the difficulty of training an imbalanced dataset. Dataset A only considers the vessel types other, fishing, tug, military, pleasure, pilot, enforcement, passenger, cargo and tanker. Dataset B only consists of the four biggest classes, which are fishing, tug, passenger and cargo. Both datasets are used to train multiple deep learning algorithms such as MLP and CNN. The results show that the ensemble of six trained models scores the highest accuracy at 75% for dataset A. The ensemble method also scored the highest accuracy at 92% for dataset B. These accuracy results are promising for the research question of this graduation thesis. However, the exact manipulation of the dataset during the pre-processing steps simplifies the reality. This cannot be implemented in this research, since during

the practical implementation on board each vessel travelling along the naval vessel should be classified and certain vessel types cannot be ignored.

5 Data

The data originates from AIS messages. AIS stands for Automatic Identification System, which is a system developed for tracking vessels. Each AIS system consists of one VHF (very high frequency) transmitter, multiple VHF receivers and standard marine electronic communication links to shipboard display and sensor systems, as is stated by the U.S. Department of Homeland Security [14]. The range is limited to the VHF range, which is about 18 to 37 kilometres. Receiving and transmitting AIS information is mandatory for international ships with 300 or more tonnage and all passenger ships regardless of the size. The AIS messages can be differentiated between several types of messages. Based on the message ID the system handles 27 different kinds of messages. Many of those are similar and do not contain valuable data for this problem. Therefore, this research focuses on data obtained from messages with message ID 1, 2, 3, 5, 9, 18, 19 and 27. The exact content of these messages can be found at the site of the U.S. Department of Homeland Security [1]. The data originating from those messages is most interesting because it contains either static and voyage related messages or position related (dynamic) messages. The dynamic messages are automatically sent every 2 to 10 seconds, depending on the speed of the vessel. In case the vessel is at anchor or moored, the messages are sent every 3 minutes. The remaining static data is updated every 6 minutes and require human involvement. A concrete description of the static and dynamic data will be specified in Section 5.1.

An AIS message is an encoded message, which is exchanged using NMEA 0183 sentences. The abbreviation NMEA stands for National Marine Electronics Association. This association has developed the interface that enables information exchange between marine electronics. The messages are structured in a certain way, which makes it possible for the receiving party to interpret the message. Such an NMEA message looks like the following.

!AIVDM,1,1,,A,14eG;o@034o8sd<L9i:a;WF>062D,0*7D

The part before the first comma indicates the message type, which represents a message received from another vessel when it is equal to !AIVDM. The second to last comma-separated part of the message, the AIS sentence, is the actual encoded AIS data using AIS-ASCII6. All other parts of the sentence are considered irrelevant for this research. After decoding the message, the desired variables can be extracted, collected into a dataset and used for modelling.

The decoding and collecting of data from the messages is implemented using the ESI AIS parser in Java. The ESI is a research institute that Thales has been cooperating with in a project involving the development of this parser. Since not all messages contain relevant data, only possible useful information is extracted from the before mentioned considered messages. The Maritime Mobile Service Identity (MMSI) number is an important variable that should be known in every message update in order to know to which vessel the information belongs. This MMSI number is a unique number of nine digits that identifies the radio system on board, amongst which the AIS. Consequently, this number usually remains the same for a certain vessel. Hence, the MMSI is used as a unique key that makes it possible to collect all updates of one vessel and to combine specific data of the concerning vessel. If the MMSI is not available in a specific update, the update cannot be linked to a vessel and is, therefore, not added to the dataset. The dataset is filled using a two-pass approach. First, all incoming message updates are processed, after which all available static data of each unique vessel is added to the corresponding vessel updates that did not contain the particular static data. The assumption that all available static information is known in every instance for each specific vessel is allowed, since external databases containing this information can be used during the real implementation on board of a naval vessel. Details about the concerning data are clarified in the next sections. The related operations will be explained in chronological order, unless stated otherwise. Each modified message update represents one instance in the dataset. Therefore, from this point forward, the message updates can be referred to as instances or observations. The final model should be able to classify each instance of the dataset.

5.1 Description

As mentioned before, a distinction is made between static and dynamic message updates. The specific data fields in the static and dynamic messages differ and are described in detail by the U.S. Coast Guard Navigation Center. Table 10.1 and 10.2 in the Appendix show these details about the dynamic and static variables, respectively. The next paragraphs explain the variables considered during this research.

The dynamic message updates contain data about the kinematics of the vessel. Each subsequent update contains a different position in case the vessel is moving. Position related messages provide valuable data fields such as the timestamp, SOG, longitude, latitude, ROT, COG, heading, navigational status and special manoeuvre indicator. Each message contains the UNIX timestamp, which can be converted into the corresponding date, hours, minutes and seconds. Note that this timestamp is different than the one stated in table 10.1 in the Appendix, which only indicates the seconds field of the UTC time when the message was transmitted. The speed over ground (SOG) represents the speed of the vessel in knots, which is set to 102.2 in case the speed is 102.2 knots or higher. The longitude is a geographical coordinate that indicates the east-west position on the Earth, and ranges from -180ř to +180ř. The latitude is the coordinate that indicates the north-south position, and ranges from -90ř to +90ř. Together, the longitude and latitude, represent the exact location of the vessel. The rate of turn (ROT) indicates how fast the vessel is turning in degrees per minute. Two quite similar kinematic values of the vessel are the course over ground (COG) and the heading. The COG indicates the actual direction the vessel is sailing relative to the true north. The heading, however, denotes the direction in which the vessel is pointing relative to the true north. The COG and heading might differ from each other in case the vessel is experiencing certain weather conditions, like current and wind. In this case the pilot needs to compensate and steer into the skid in order to keep on the right course. The navigational status is a category manually set by a ship crew member on board of the concerning vessel, and indicates the vessels current type of activity. Possible activities are under way using engine, at anchor, not under command, restricted manoeuvrability and moored. The last dynamic information that might be relevant during this research is the special manoeuvre variable, which denotes whether or not the vessel is engaged in a special manoeuvre.

The static and voyage related messages contain data about the vessel that does not change during one trajectory of the vessel. These static message updates provide data fields such as the IMO number, call sign, name, draught, destination and vessel type. The IMO number is the International Maritime Organisation number, which is a unique reference of the vessel consisting of seven digits. The call sign is a unique designation for the transmitting radio, which consists of ASCII characters. The name of the vessel is specified in a maximum of 20 ASCII characters. The draught of the vessel is represented in meters and is set to 25.5 if the draught is 25.5 meters or higher. The destination of the vessel indicates the desired destination of the vessels trajectory and consists of a maximum of 20 ASCII characters. The final two variables are derived from the vessel type number, which the AIS message contains. This number consists of two digits. The first digit represents the general category of the vessel, which will be referred to as grouped vessel type. The grouped vessel type distinguishes the categories reserved for future use, wing in ground, special, high-speed craft, passenger, cargo, tanker and other. Note that the special category consists of vessels originating from two different first digit numbers. Digit number 3 includes vessels like fishing, towing, sailing and underwater operations, while digit number 5 includes vessels like pilot, search and rescue, tugs and law enforcement. Considering the second digit along with the first digit, the vessels are divided into more sub types. These types will be referred to as the actual vessel type, and the associated text can be retrieved from the message with the ESI AIS parser. Clarification about the exact sub types is included in Table 10.3 in the Appendix.

The static related messages require human intervention before the messages are transmitted. Therefore, this data is sensitive to human error and not regarded as highly reliable. Especially the vessel type is prone to human error, including deliberately entering the wrong information. Vessels with a distorted vessel type and, therefore, a temporary false identity are guilty of spoofing. Particularly the vessel type is a variable that is essential to be correct, since it is the target variable. The vessel type variable is, therefore, validated in Section X to create ground truth target labels. Especially for special operations of the Dutch Navy, it is important to act upon secure and trustworthy information. Such operations are more likely to get involved with dangerous situations, in which case the highest possible level of transparency is needed in order to make wise decisions and act accordingly. Therefore, the AIS data needs to be checked on threats like deception and disruption. That is, a trustworthy model will be achieved when the data is

FIGURE 5.1: Area of the vessels

trustworthy. However, this check is not implemented in this research as it is not within the scope. The AIS data, apart from the vessel type, is from this point further assumed to be correct. The remaining static variables can be assumed with reasonable certainty to be correct, because the data originates from the North Sea. In general, vessels in this area can be considered reliable. Although this assumption is made, special attention for the static variables is appropriate during data analysis and feature selection. The incoming dynamic and continuous AIS variables of this dataset can be considered reliable and accurate with certainty, since those variables are checked and compared in the CMS with data originating from the sensors on board of the naval vessel.

5.2 Analysis

This exploratory data analysis consists of a descriptive analysis, such that useful insights and patterns in the data are discovered and clearly presented. The analysis results will add value during future decisions in the machine learning process. The performance of models obtained by machine learning algorithms is most dependent on the data used during training the algorithm. Therefore, understanding the data is essential in order to effectively pre-process the dataset. Part of this understanding is learning more about the context of the data, which is needed to make well-informed decisions that results in a good model performance. First, some general information retrieved from the data will be presented. Subsequently, the missing values of all variables are analysed. The explanatory variables can be divided into categorical and continuous variables, each of which require different analysis methods, like histograms and boxplots, to get a clear understanding. The target variable will be extensively analysed. The composition of the vessel types is required to identify possible dataset imbalance. Finally, the analysis includes tests to discover relations between all variables.

Some general analysis shows that the AIS messages cover a time period of 41 minutes and 8 seconds, with the first timestamp being equal to the date 11^{th} of October 2006 and time 15:39:53. During this time period, the message updates of 753 unique MMSI numbers are collected. The total number of updates and, therefore, the number of instances in the dataset is equal to 68,340. This means that the average number of updates per vessel is equal to approximately 91 updates, with a minimum number of updates of 1 and a maximum of 2027 updates. The geographical location of these 753 vessels is the North Sea, including parts of the internal waters of the Netherlands and Belgium. A large majority of the vessels is located in the port of Rotterdam. The concerning area is displayed in Figure 5.1.

Possible missing values in each variable need to be identified, since missing values effect the results of the other methods used in the analysis and lead to certain decisions. Missing values represent incomplete data and are unwanted, because they preclude total transparency of the appearing circumstances and

consequently make the learning process of the machine learning algorithms harder. The percentage of missing values in the instances of each variable is shown in Table 5.1. High percentages of missing values occur for the variables *ROT, heading, draught* and *destination*. The variables *IMO, call sign, name* and *vessel type* are missing a significant amount of values as well. How these missing values are dealt with will be discussed in Section 5.3.3. The zero missing values for the variable *grouped vessel type* is rather remarkable, since the variable *vessel type* does have missing values. However, this is caused by the internal programming decisions, which have led to assigning the missing values of this variable to the 'other' category.

TABLE 5.1: Missing values

Variable	Percentage (%)	Mean percentage per MMSI (%)	Number of MMSI's	Mean percentage per MMSI | MMSI with missing (%)	Imputation method
MMSI	0	0	0	-	-
Timestamp	0	0	0	-	-
SOG	0.9994	0.7440	9	62.2438	Interpolation*
Longitude	0.3541	0.3450	6	43.2985	Interpolation
Latitude	0.3541	0.3450	6	43.2985	Interpolation
Navigational status	0.0571	0.1328	1	100	Mode (adjusted)
ROT	15.6760	14.9319	115	97.7715	-
COG	0.9994	0.7440	9	62.2438	Interpolation*
Heading	14.7747	13.0236	99	99.0585	Interpolation
IMO	8.6860	18.0611	136	100	-
Call sign	9.1542	18.3267	138	100	-
Name	8.6289	17.9283	135	100	-
Draught	17.8124	23.3732	176	100	Median
Destination	17.0208	24.7012	186	100	-
Special Manoeuvre	0.0571	0.1328	1	100	Mode
Vessel type	8.6289	17.9283	135	100	-
Grouped vessel type	0	0	0	-	-

* Extra calculations based on a priori data are considered when all feature instances are unknown for a unique vessel. More details are described in Section 5.3.3

The continuous explanatory variables will be examined using histograms and boxplots. For both analysis methods, the instances with missing values for each concerning feature cannot be included. Both analysis methods are convenient for displaying the distribution of a specific continuous variable. A histogram is a representation of the frequency distribution of a variable, which is realised by grouping the observations into predetermined bins. It serves as estimate for the actual underlying probability distribution. A boxplot visually represents the distribution of the variable based on five summary numbers, which are the 'minimum', first quartile, median, third quartile and 'maximum'. How these numbers are defined is shown in Figure 10.3 in the Appendix. Note that for this analysis method, the minimum is differently defined than the actual smallest observation value of the set. Similarly, the maximum is not defined as the actual highest observation value of the set. Observations outside this range are called outliers. However, these observations do not automatically need to be removed from the dataset. Instead outliers need special attention and consideration. This will be discussed in Section 5.3.5. The continuous explanatory variables that will be analysed using these two methods are the *SOG, ROT, COG, heading* and *draught*. For interpretability reasons, simply the plots for the variables *SOG* and *draught* are considered here, since these variables show most significant results. The boxplots and histograms are shown in Figures 5.2, 5.3, 5.4 and 5.5 for the variables *SOG* and *draught*, respectively.

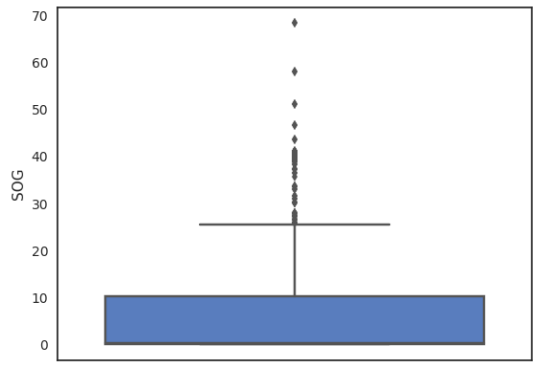

FIGURE 5.2: Boxplot SOG

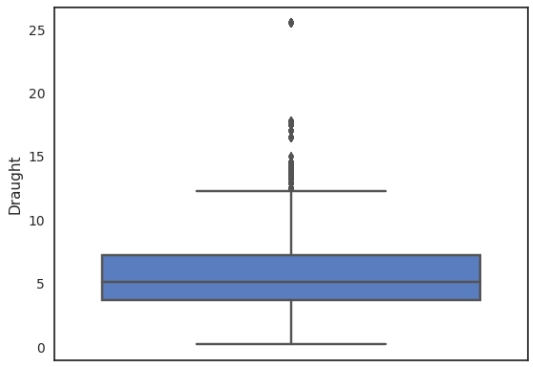

FIGURE 5.3: Boxplot draught

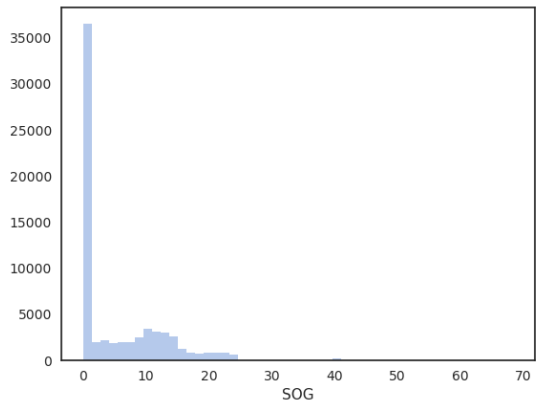

FIGURE 5.4: Histogram SOG

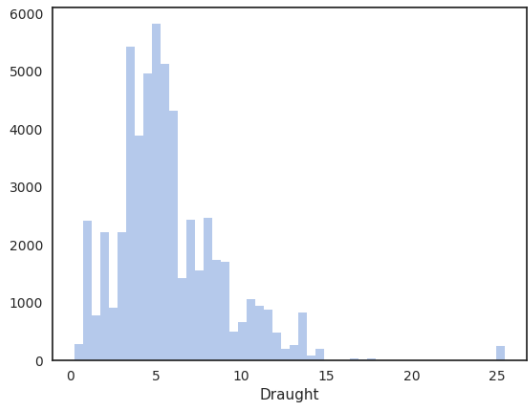

FIGURE 5.5: Histogram draught

These four plots illustrate the overall distribution of the two variables. It can be concluded that the variable *SOG* is not symmetric, because of the many vessel observations with a speed equal to 0. This skewed distribution causes the boxplot to represent many outliers, for the higher value observations. How these observations will be considered during the rest of this research is described in Section 5.3.5. The variable *draught* is rather symmetric with the exception of the outliers determined in the boxplot. However, the histogram shows that only some observations with a draught of 25.5 meters deviate a lot from the other observations, which is caused by the presence of one vessel. Each of the continuous explanatory variables can also be plotted in a boxplot for each grouped vessel type category individually, in order to provide an indication of the discriminating capability of a specific variable towards the grouped vessel type. The relevant continuous variables per grouped vessel type are, again, the *SOG* and *draught*. These plots are shown in Figures 5.6 and 5.7, respectively.

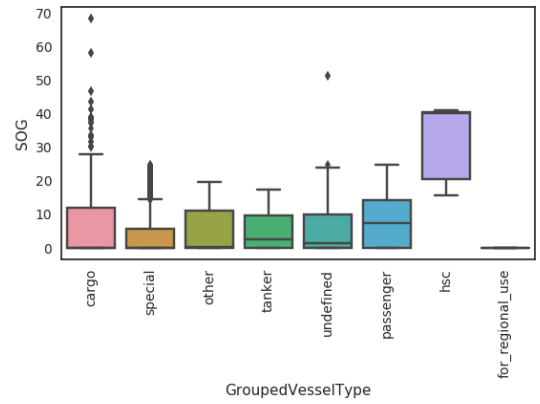

FIGURE 5.6: Boxplot SOG per grouped vessel type

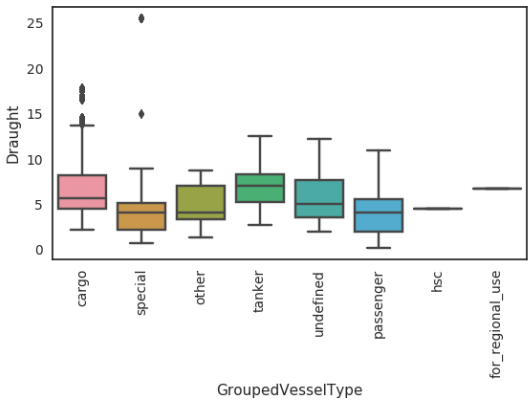

FIGURE 5.7: Boxplot draught per grouped vessel type

Especially the draught seems to have visually different distributions for some grouped vessel type categories. However, the discriminating capability is not significant in order to classify the grouped vessel type only based on the draught. The variable *SOG* shows a distinctive pattern for the category 'HSC', which has a significant higher speed compared to the other categories. This is supported by domain knowledge about the vessel types, since the category high-speed crafts (HSC) are designed for speeds up to 42 knots. The category 'cargo' also deviates from the other categories, since the distribution contains more outliers for the relatively high-speed values. Because of the high percentage of speed values equal to 0, these boxplots might lead to a distorted image. Therefore, the same boxplot analysis has been performed for the dataset without observations with a *SOG* value equal to 0. These results are shown in Figure 5.8 and indicate small changes in the medians of the variables.

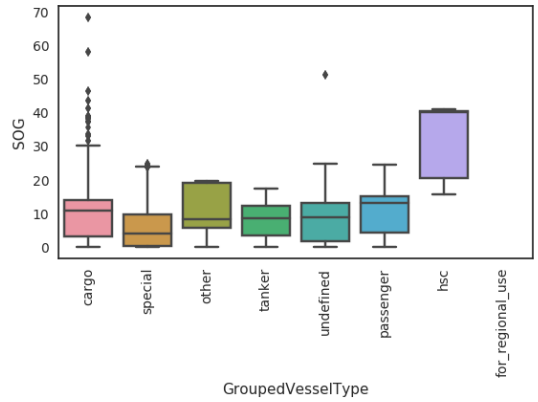

FIGURE 5.8: Boxplot SOG per grouped vessel type without observations of SOG = 0

The categorical explanatory variables can be examined using bar charts. A bar chart represents the proportions of each specific variable category in a plot. Observations with missing values are not considered in such a representation, since those cannot be assigned to a category. The categorical explanatory variables that can be analysed using a bar chart are the *navigational status* and *special manoeuvre*. Simply the plot for the variable *navigational status* is shown here in Figure 5.9. The bar chart belonging to the variable *special manoeuvre* does not add significant value, since a numerical analysis is enough to get a clear impression of the variable distribution. The majority, i.e. 70.6%, of the instances are categorised as 'no special manoeuvre indicator'.

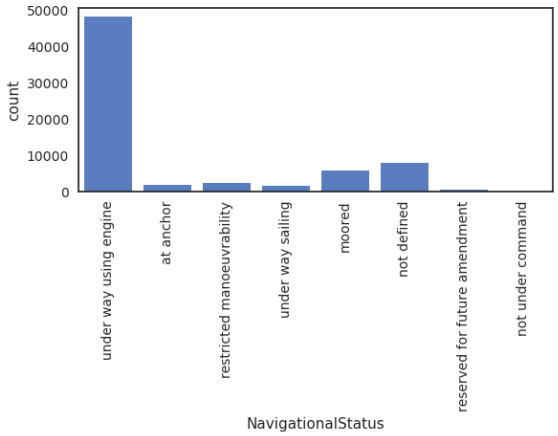

FIGURE 5.9: Bar chart navigational status

The bar chart of the *navigational status* in Figure 5.9 shows that most vessels transmit the status of being 'under way using engine'. Very few observations have the navigational status of 'reserved for future amendment' or 'not under command'. Note that this dataset does not contain all possible categories of the variable *navigational status*. This dataset contains eight out of the total of sixteen categories. Dividing the eight categories into the corresponding *grouped vessel type* does not give significant clarifying results and is, therefore, not included in this analysis.

The target variable will be analysed using histograms. The number of unique sub vessel types in this dataset is equal to 45. The list containing these 45 categories is attached in the Appendix in Table 10.4, containing a conversion of the sub types into numbers because of better interpretation of the histogram. These categories are plotted in the histogram in Figure 5.10 and clearly show the inequality of vessel type observations and identify a majority of category 5. Number 5 indicates a majority of cargo vessels, which is representative for the worldwide situation. The histogram in Figure 5.10 visualises the number

of observations, which might differ from histogram of the number of unique vessels. Therefore, this histogram is shown in Figure 5.11.

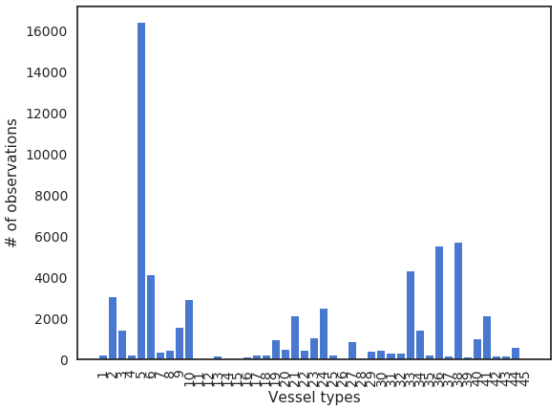

FIGURE 5.10: Histogram of vessel types (observations)

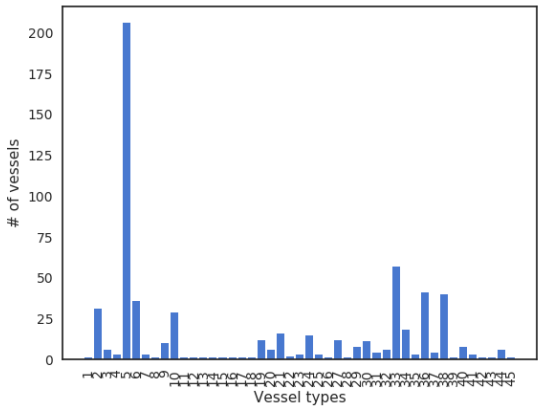

FIGURE 5.11: Histogram of vessel types (vessels)

Note the small differences between the histogram for the amount of observations and the histogram for the amount of vessels with the corresponding vessel type. No significant changes in the distribution of the vessel types indicates that it is allowed to continue using the observations during the analysis. The inequality in observations of the vessel type categories will be further elaborated in Section 5.3.6, containing what factors need attention during transforming the variable vessel type into the final target variable. One option would be using the grouped vessel type as target variable. The corresponding histogram is displayed in Figure 5.12.

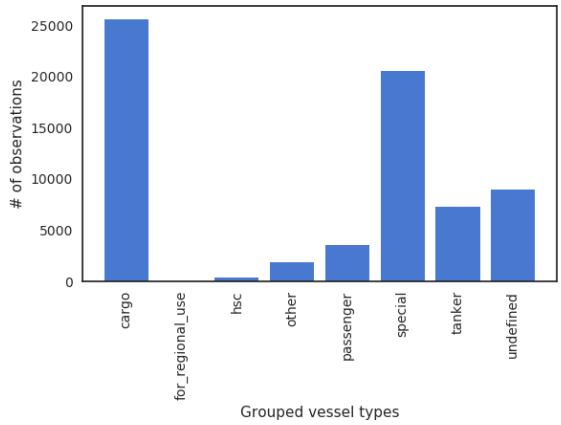

FIGURE 5.12: Histogram of grouped vessel type (observations)

The histogram in Figure 5.12 shows the cargo category to be the most occurring category, which is in accordance with the normal vessel type histogram shown in Figure 5.10. Moreover, the categories are still significantly unevenly distributed.

Finally, the relationship between all variables will be analysed. The used methods cannot deal with missing values, which means only pairwise complete observations will be considered. First, the Pearson correlation between continuous values will be computed. The Pearson correlation coefficient $\rho_{X,Y}$ is a measure between two variables X and Y, which computes their linear relationship. It is calculated as follows.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \in [-1,1], \tag{5.1}$$

where $cov(X, Y)$ is the covariance between the variables X and Y, σ_X is the standard deviation of the variable X and σ_Y is the standard deviation of the variable Y. A value of 1 indicates total positive linear correlation, 0 indicates no linear correlation and -1 indicates total negative linear correlation. Besides the Pearson correlation, also the Spearmans rank correlation is considered here. The Spearmans rank correlation coefficient r_s measures the monotonic relationship between two variables X and Y. It computes the Pearson correlation coefficient between the ranked variables, which is calculated as follows.

$$r_s = \rho_{rg_X, rg_Y} = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}} \in [-1, 1], \tag{5.2}$$

where ρ_{rg_X, rg_Y} indicates the Pearson correlation coefficient applied to the ranked variables, $cov(rg_X, rg_Y)$ is the covariance of the ranked variables, σ_{rg_X} is the standard deviation of the ranked variable X and σ_{rg_Y} is the standard deviation of the ranked variable Y. The resulting Spearmans rank correlation coefficient is similar to the interpretation of the Pearson correlation coefficient. More specifically, a Spearmans correlation of 1 results when all observations with higher values for X than a certain observation will also have higher values for Y. This means the Spearmans rank correlation coefficient is able to identify more relationships between two variables. The pairwise Pearson correlations and pairwise Spearmans rank correlations are computed and shown in Table 10.5 and 10.6 in the Appendix, respectively. Both methods only show significant correlation between the variables longitude and latitude. This relation is supported by domain knowledge, since certain longitude and latitudes combinations might indicate the vessel being in a specific area. Moreover, little correlation is found between the variables heading and COG, which will be further analysed. The remaining pairwise correlation coefficients are not significant. However, this does not exclude the possibility of a more complicated correlation between two other variables. Since predicting the vessel type is the aim of this research, the relation between the explanatory variables and this variable is most interesting to analyse. Once a clear relation exists between a variable and the target variable, this means the explanatory variable is useful and has good predictive power. However, the Pearson correlation coefficient and Spearmans rank correlation coefficient cannot be used, since the target variable is categorical instead of continuous. A reasonable solution is the approach of plotting the continuous variables pairwise with different colours for the different grouped vessel types. This will give a rough indication of the discriminating capability of the two continuous variables towards the target variable vessel type. Note, these plots are made for a subset of the whole dataset, all instances with a missing value for at least one of the continuous variables are not considered in the pairplots shown in Figure 5.13.

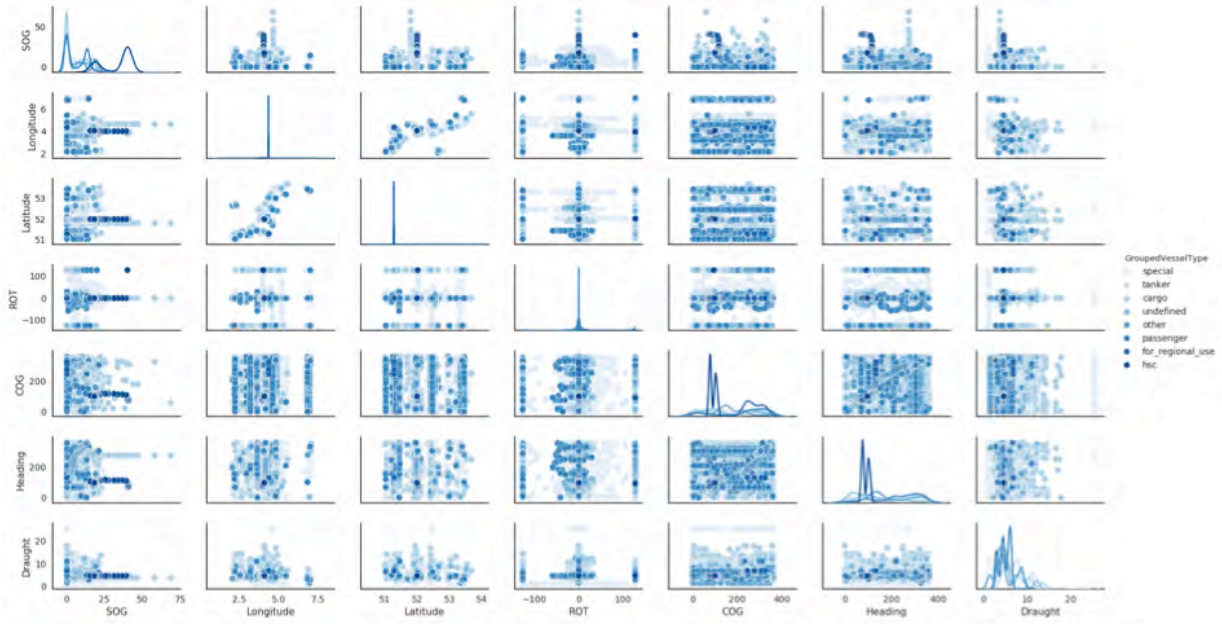

FIGURE 5.13: Pairplots of continuous variables

These correlations are analysed with the domain knowledge of a human operator. High correlations are expected between the variables *longitude* and *latitude*, *COG* and *heading*, and *longitude* with *latitude* and

SOG. The human operator has indicated more relations between certain variables or combination of variables, but these cannot be derived from the pairplots. The correlation between *longitude* and *latitude* is clearly visible and supported with the operators knowledge. The combination of longitude and latitude indicates the exact location of a vessel. Some important information within these values is whether or not a vessel is located at a certain area, like a shipping lane or a harbour. Human operators on board of a naval vessel use this information in combination with the vessels speed to determine the type of vessel. This also motivates the expected correlation between *longitude*, *latitude* and *SOG*. However, this cannot be derived from pairplots. Therefore, the feature 'area' will be engineered in Section 5.3.4. The remaining correlation that can be derived from the pairplots is between the variables *COG* and *heading*. At first sight the plot shows observations divided over the whole field. However, the graph also shows a clear linear line, which is because the COG and heading are often similar. The deviating observations from this linear line can be caused by the weather conditions that make the two variables deviate from each other for certain vessel trajectories. Especially the Pearson correlation is high for certain grouped vessel types. The Pearson correlation for a tanker is equal to 0.698405, for a passenger equal to 0.668424 and for an HSC equal to 0.999378. Because of this high correlation and similar meaning of the variables, one of the variables should not be included as predictive feature in the model. This, because using both means one of them is redundant. All other continuous variables do not show significant correlation, which means they could be used together as input in the machine learning algorithms.

Besides the approach of the pairplots, two different kind of tests can be executed in order to test whether an explanatory variable is independent from the target variable. Once independence is indicated for a certain variable, this means that there is no need to consider it as variable to predict the vessel type. The first test is for testing a continuous explanatory variable towards the categorical target variable, while the second test considers a categorical explanatory variable compared to the categorical target variable. The one-way ANOVA test determines whether the mean values of the different independent groups are significantly different. This means it tests the following null hypothesis H_0 for k independent groups against the following alternative hypothesis H_1.

$H_0 : \mu_1 = \mu_2 = ... = \mu_k$,
H_1: not all mean values of the different groups are equal,

where μ_i is the mean of group i. This null hypothesis will be rejected and the alternative hypothesis will be accepted in case the p-value is equal to or smaller than 0.05. The one-way ANOVA test is performed for the continuous features *SOG, ROT, COG, heading* and *draught* in combination with the grouped vessel type. The resulting p-values for the features *SOG, heading* and *draught* are equal to 0 and the p-values of the remaining two variables are approximately 0. This indicates a rejection of the null hypothesis for all features, which means that for each variable at least the mean values of two vessel type groups are significantly different. The target variable is, therefore, not independent from these five considered continuous variables. Consequently, these variables are relevant to consider during feature selection. A reasonably good visual indicator of this test is a boxplot such as in Figure 5.6, which represents the specific variable distribution for each grouped vessel type. A Chi-square test determines whether two categorical variables, with both mutually exclusive categories, are independent. This means it tests the following null hypothesis H_0 against the following alternative hypothesis H_1.

H_0: two categorical variables are independent,
H_1: two categorical variables are dependent.

This null hypothesis will be rejected and the alternative hypothesis will be accepted in case the p-value is equal to or smaller than 0.05. The Chi-square test is performed for the categorical features *navigational status* and *special manoeuvre* in combination with the *grouped vessel type*. The resulting p-values for both features are equal to 0. This indicates a rejection of the null hypothesis for both categorical features, which means that both features are dependent with the grouped vessel type. Consequently, these variables are relevant to consider during feature selection. Note, for both features a big majority of the vessels belong to one specific type of feature category. These previous executed methods give an indication about the relation between the explanatory variables and the target variable. Additional methods are considered during the feature selection, described in Section 5.3.7, in order to examine the predictive power of the explanatory variables.

5.3 Pre-processing

The quality of the data is the most important aspect for providing a high performance. If data quality is low, the algorithms are not able to find the existing patterns in the data between the explanatory variables and the target variable. Defining data quality is rather hard, but here the definition of Apon et al. (2017) [10] is used. 'Data quality is defined in terms of its fit for a business purpose.' However, this definition is qualitative and does not indicate a quantitative form of measuring data quality. Furthermore, data quality assessment is domain specific, involves a considerable degree of subjectivity and requires significant human involvement. In order to make this more concrete, the lack of quality is often determined by factors such as data that is inaccurate, inconsistent, incomplete, unreliable, irrelevant and dated. The factors that are essential to this research were considered in the previous data sections.

The previous two sections reflect the data quality challenges within this research problem. Typical and appropriate data challenges in this research are dealing with missing data, duplicate data, data volume, outliers, unreliable data and imbalanced data. All these factors need to be considered during the pre-processing of the data, in order to optimise the quality of the dataset. The quality of the data might also be algorithm specific, since particular algorithms are sensitive for certain characteristics. In this respect, data transformation and feature selection are implemented in order to increase the quality and, consequently, increase the functioning of a specific algorithm. The final aspect that is part of the pre-processing of the data is feature engineering. This involves adding extra features to the dataset in order to enlarge the data quality by additional information or derived information.

Since the quality of the data is of big importance, the data pre-processing is considered as most important activity during this research. A pre-processed dataset has a direct effect in a more efficiently training process of the machine learning algorithms. Especially applying machine learning in the naval domain requires good pre-processing, at least in case AIS data is used. Main reasons are the possible (human) errors, incomplete data fields and necessary general subdivision of the vessel types. However, it is crucial to pay attention to over-processing the data. Actions like removing specific instances or handling missing values in a certain way may result in adding too much bias to the data. It seems to improve the data quality based on domain knowledge, while adding too much domain knowledge actually reduces the generalisation of the data. A biased dataset is prone to overfitting, which causes a model to only be able to give good predictions for the concerning dataset. Adding little bias to the data is acceptable, as long as the implementation is deliberate and the consequences are known and dealt with.

This section describes the actions executed to prepare the dataset before using it as input for training the machine learning algorithms. Pre-processing this dataset covers removing duplicates, splitting the set into a training and test set, dealing with missing values, removing outliers, engineering features, transforming features, creating target labels and selecting features. The following subsections discuss the implementation of these processes.

5.3.1 Duplicates

The first step in pre-processing the dataset is handling the duplicate instances. Duplicate instances indicate message updates that are received at least twice. A duplicate for this dataset is defined as an instance that has the same values for the variables *timestamp* and *MMSI* compared to another instance. Identifying and eliminating duplicate instances is critical, since this will better represent the real world circumstances. A duplicate message is caused by some kind of transmission error and does not add extra and reliable information to the dataset. Here it is assumed that receiving two different messages at the same time for the same MMSI is impossible, because a vessel cannot be at two different locations at once. Therefore, only the messages with successive timestamps for a specific MMSI number will be maintained. This means only the first unique instance combination of *timestamp* and *MMSI* remains in the dataset, while the duplicates are removed. This first instance cannot deviate much in terms of location and speed from duplicate instances, which has led to keeping the first instance for applicability reasons. This process is implemented before performing the data analysis, since the processes in reverse order may lead to skewed observations during data analysis. The reason for the skewed observations in the analysis is that the duplicates will count for multiple times. Removing the duplicates caused the dataset to decrease from 72,179 to 68,340 message updates.

It should be noted that the previously mentioned assumption is debatable. The dataset contains instances with the same *timestamp* and *MMSI* number combination as another instance, yet have different values

for the *SOG*, *longitude* and *latitude* variables. This indicates that two updates occurred at the exact same time, both stating that the vessel was sailing with different speed at a different location. This is impossible and will cause unusual values for the features that will be engineered during feature engineering. Calculating the acceleration between two of those instances will result in an undetermined number, since the denominator is equal to 0. However, the speed, longitude and latitude combination indicate movement between the instances, which cannot be incorporated into the dataset. A reasonable effect is irregularity in the acceleration sequence of the vessel that did not occur in reality. Consequently, the best option appears to remove duplicates based on unique combinations of *timestamp* and *MMSI* number, which is what is implemented.

5.3.2 Data splitting

Before additional data pre-processing will be performed, the dataset is split in a training set and test set of approximately 80% and 20% of the original dataset, respectively. These percentages are common division figures, especially when part of the training set will be used for validation. The validation set is useful for validating the intermediate performance results. The data splitting is necessary before the imputation of missing values, considering that the training set is not allowed to contain information from the test set because it makes the training set biased. This obligatory independence between the training and test set determines the required specific kind of splitting method. The process is not allowed to randomly select 80% of the message updates (instances) and consider these as the training set, because all instances of each unique MMSI number should be together in either the training or the test set. The test set is required to be independent from the training set, since it should represent a real-life situation. In such circumstances the training set does not contain the message updates of the same vessel track as the message updates that will be used when implementing the model to predict newly seen track updates. Since all message updates of a unique vessel track are dependent, these are either split into the training set or the test set. Therefore, around 80% unique MMSI numbers are selected first. All corresponding instances of those MMSI numbers together form the training set, which leaves the remaining instances belonging to the test set.

5.3.3 Missing values

Imputation of missing values is a process that is rather avoided, since the imputed value is never completely certain. It may result in a biased dataset. Imputation is not necessary for all machine learning algorithms. However, this research does require the handling of missing values, given the chosen Scikit-learn library that is not able to cope with missing data in all its algorithms. It is, therefore, a requisite to replace the missing values with an estimate. Moreover, it is important to perform the missing values imputation before implementing feature engineering, since all values are required to be available when engineering features based on the existing features. This implies that missing values cannot be imputed with values derived from engineered features, hence a priori data and useful external data are considered appropriate for imputing missing values. External databases, like databases originating from marinetraffic.com or digital-seas.com, contain facts about many vessels derived by retrieving information about the specific MMSI number. Vessel specific static information within these databases are appropriate for imputing the corresponding missing values. However, these databases are not publicly available, which means the required data needs to be bought or obtained manually. Due to no available money and time considerations it is decided to not use these databases. Which imputation method will be implemented for each feature is supported by domain knowledge of a human operator. Such an imputed value comes closest to the real value of the variable. An overview of which imputation method is used for each feature is shown in Table 5.1. Each method is described in the remaining paragraphs of this section.

High percentages of missing data require careful attention, regardless of the cause. Several approaches exist for dealing with missing values. The simplest approach is to remove all instances with missing values from the dataset. However, applying this approach to this specific research will result in an even smaller dataset, which might have lost potentially important information. The model is simply required to be able to deal with imputed values, since incoming messages containing missing values cannot be ignored. Those corresponding instances need to be classified by the implemented model too. Moreover, if a critical discriminating variable consists of a large proportion of instances with missing values, its removal will have an effect on the statistical power [10]. These statements considered together indicate that the missing values of the explanatory variables should be imputed.

A preferable procedure to impute missing values is regression imputation. Regression imputation is based on the values of other variables within the dataset. A regression model will be trained to predict the values of the considered variable based on the other variables. The missing values are then imputed with the predicted values from the regression model. It should be noted that all values of the other variables must be available in order to apply the model. Moreover, this imputation approach requires the dataset to contain some features that are highly correlated with the concerning feature. For example, if there would be a significant relation between the variables *draught, longitude* and *latitude* and the *speed*, these three variables could be used to determine the missing values occurring in the variable *speed*. Since this requirement does not sufficiently fit any combination of variables in the considering dataset, regression imputation is not used in this research.

Another approach to impute missing values is to substitute them with a mean, median or mode value. The mean and median can only be considered when the feature is continuous, whereas the mode can also be used when the feature is categorical. The instances with missing values for the feature *draught* are substituted with the median. This means all missing values are assigned with the median value computed based on all available values of its corresponding vessel type. Since the draught is missing for all instances of certain vessel types, there are still missing values after the first imputation. Therefore, the remaining missing values are imputed with the median value computed based on all available values of its corresponding grouped vessel type. This imputation approach can be applied for the training set, since the vessel type is known. However, for imputing the missing values in the test set, the vessel type is the prediction target and cannot be used for the same imputation approach. This means the median of all available instances should be used. If the total number of available instance values is an odd number, the median is computed as follows.

$$\text{Median} = (\tfrac{n+1}{2})^{th}\ instance\ value,$$

where n is the number of instances in the set. If the total number of available instance values is an even number, the median is computed as follows.

$$\text{Median} = \frac{(\frac{n}{2})^{th}\ instance\ value + (\frac{n}{2}+1)^{th}\ instance\ value}{2}$$

where n is the number of instances in the set. The mode substitution approach is considered for the feature *special manoeuvre*, since the feature is categorical. The instances with missing values are substituted with the most frequently occurring category of the available concerning values. This implementation suits this feature, since there is a strong probability that the missing values actually are equal to the mode value. Reason for this is the fact that the mode of special manoeuvre covers 98.68% of the available instances and very few instances have a missing value for this feature. The disadvantage of the mean/median/mode substitution approach is the reduction of data variability. An advantage, however, is the maintenance of the sample size.

The other categorical feature having its missing values imputed is *navigational status*. The imputation approach has an additional check, because a priori data about the speed is used besides substituting the mode. Once the speed of the vessel is smaller than 0.1, the missing value of *navigational status* is set to 'moored'. Otherwise, the missing value is substituted with the mode, which is equal to 'under way using engine' and covers 70.64% of the available instances.

An accurate imputation approach is interpolation. Because of the continuous nature of the following features, interpolation can be used. The features *longitude* and *latitude* are required to determine other features and are, consequently, interpolated using the spline interpolation method. After applying this method the first feature value of a vessel might still be missing. This value will be imputed with the value of the next vessel instance. If all instance values of the concerning feature are missing for a specific vessel, no adequate method is available to impute those. Especially the *longitude* and *latitude* values should be certain and the instances that still have missing values after the imputation method are, therefore, deleted from the dataset. The features *SOG, COG* and *heading* all are linearly interpolated for each unique MMSI number. In some cases the first instance is still missing, in which case it will be imputed with the value of the next instance. For all three features the dataset contains vessels with missing values for all instances, in which case a priori data of other features might be used to derive the missing values. Whenever all instances of a certain MMSI are missing for the feature *heading*, the missing values are imputed with the *COG* of the concerning instance. For the feature *SOG*, the value will be derived from the *longitude, latitude*

and *timestamp* of the current and previous instance, as is shown in the following equations.

$$\begin{gathered} SOG = \frac{distance}{\Delta t} \cdot \frac{1}{0.514444}, \\ distance = R \cdot c, \\ c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{(1-a)}), \\ a = \sin^2\left(\frac{f(lat_2 - lat_1)}{2}\right) + \cos(f(lat_1)) \cdot \cos(f(lat_2)) \cdot \sin^2\left(\frac{f(long_2 - long_1)}{2}\right), \\ R = 6371000, \\ f(x) = x \cdot \frac{\pi}{180}, \end{gathered} \tag{5.3}$$

where $distance$ is the distance between the two instances in metres, Δt is the time difference in seconds, SOG is the speed over ground in knots, $long_1$ and $long_2$ are the longitude values for the two instances, $lat1$ and $lat2$ are the latitude values for the two instances, R is the earths radius in metres and $f(x)$ is the function that transforms degrees to radian. The first instance value of a specific vessel is an exception to this formula, since it does not have a previous value. It will be assigned with the value of the next instance. Regarding the missing values of the feature *COG*, the values will be set to 0 in case the mean of the *SOG* values is smaller than 0.1. This, because it indicates a vessel that is not sailing, which means the course over ground is irrelevant. Otherwise, in case the mean of the *SOG* is equal or bigger than 0.1, the value will be derived from the *longitude* and *latitude* of the current and previous instance, as is shown in the following equations.

$$\begin{gathered} COG = (\theta + 360) \,\%\, 360, \\ \theta = g(\text{atan2}(y, x)), \\ y = \sin(long_2 - long_1) \cdot \cos(lat_2), \\ x = \cos(lat_1) \cdot \sin(lat_2) - \sin(lat_1) \cdot \cos(lat_2) \cdot \cos(long_2 - long_1), \\ g(x) = x \cdot \frac{180}{\pi}, \end{gathered} \tag{5.4}$$

where COG is the course over ground in degrees, % is modulo, $long_1$ and $long_2$ are the longitude values for the two instances, lat_1 and lat_2 are the latitude values for the two instances and $g(x)$ is the function that transforms radian to degrees. The first instance value of a specific vessel is an exception to this formula, since it does not have a previous value. It will be assigned with the value of the next instance.

The missing values for the remaining features *IMO, destination, call sign, name* and *ROT* are not imputed. The categorical variable destination does contain valuable information, but requires manual cleaning since the data is not consistent in its categories. Based on domain knowledge is decided that the remaining three categorical feature values do not contain enough predictive power to predict the vessel type. The *IMO, call sign* and *name* are too specific for a unique vessel instead of related to a certain vessel type. Therefore, it is not valuable to impute the missing values. The features can still contain some value, which is considered during feature engineering in the following subsection 5.3.4. The feature *ROT* is excluded from the dataset, since it contains a large proportion of missing values and imputing those with a mean or median has no added value. The missing values could be derived from other feature values. However, this takes much effort compared to the expected value it will add to the predictive power of the model.

An exception is made for dealing with the missing values of the response variable *vessel type*. Since it is the target feature of the model, these feature values need to be certain. Building a model to predict this feature is precisely the problem. Hence, the instances without the vessel type are not imputed but removed from the dataset. In particular, due to the two-pass approach, this involves removing all instances of certain unique MMSI numbers. The remaining dataset still consists of 618 unique MMSI numbers of the original 753 unique vessels. As an additional result, all instances with missing values in the variable *name* are removed as well, which leaves no missing values for this variable.

5.3.4 Feature engineering

Feature engineering is a process within machine learning that is implemented with the aim to increase the predictive power of the model. Possible valuable information will be added to enrich the dataset by

creating features with empirical knowledge. This additional data can be derived from existing features in the dataset. For example, combining several feature values might be useful or deriving information from several instances of one specific feature. Alternatively, features can also be engineered based on external sources. For example, the databases behind marinetraffic.com and digital-seas.com contain additional general information, like the gross tonnage, about many vessels. However, external sources will not be used, since such databases are not available internally and manually looking it up on the Internet took too much time.

The features are designed based on domain knowledge, literature study and the results of the correlation analysis. Most domain knowledge is acquired through interviews with an ex-operator, who has spent significant time on board of a naval vessel. An operator acquires a vast amount of knowledge through practical experience. The hands-on experience provides understanding of what is important in order to classify the vessel manually. The most significant factor for predicting the vessel type manually is the movement history of the vessel, which will be reflected in most of the engineered features. The dataset consists of temporal sequences of each vessel. This classification problem is not considered as a time series, but it is preferable to add the valuable temporal information. This temporal information is, therefore, translated into the new features containing the history of the vessel. Other domain knowledge originates from previous internal researches and indicates the type of area as a seriously decisive factor. Related literature shows what features have been relevant in those researches. If other researches have significant similarity to this research, those engineered features might add value to this classification problem. Therefore, these features or variations of these features are considered. Relevant feature engineering ideas are derived from Cheng et al. (2017) [24] and Mascaro et al. (2010) [20]. Finally, correlation analysis combined with domain knowledge can indicate what features combined are more useful than those features considered individually.

As evidenced by hands-on experience and previous internal research, the type of area the vessel is located in, in combination with the sailing behaviour of the vessel, has significant impact on determining the vessel type. Therefore, the feature area type will be engineered and the behaviour of the vessel needs to be incorporated in other features about its speed and sailing direction. Four different area types are distinguished, which are the areas harbour, fishing, shipping and anchor. Distinction is made between these areas because each area entails a different kind of common sailing behaviour. In a harbour area the vessels are either moored or sailing with the same maximum speed. The vessels in an anchor area are not sailing at all. Vessels in a shipping area are generally sailing with its economic speed, which is financially the most favourable speed due to consuming the least amount of fuel. And the remaining vessels are distinguished as located in fishing area, since the remaining area theoretically is fishing area. Precise delimitation of the areas is essential for the model performance. A small shift of area coordinates can have impact on assigning the appropriate area to an instance. Once a wrong area type is assigned, the machine learning algorithm might learn incorrect patterns which will result in a poor model performance. The anchor areas are defined using the live map of marinetraffic.com, which shows the current worldwide vessel composition. Anchor areas are easily distinguished into five areas, since the vessel speed is equal to 0 and many vessels are near each other. The harbour area is defined as the harbours of Den Helder, Rotterdam, Amsterdam, IJmuiden and the entire coastline. Shipping areas are determined by the two shipping lanes located on the North Sea. The border coordinates of each area are used to determine the polygon representing the right area. In which area each vessel is located in is determined by its longitude and latitude in the corresponding message update. If the longitude and latitude combination is within one of the three predefined areas, the concerning area will be assigned to the instance. Otherwise, the vessel is considered to be in a fishing area. The specified areas are shown in Figure 5.14.

Many of the remaining engineered features are incorporated into the dataset to represent the vessel behaviour. The behaviour of a unique vessel is best represented by taking into account the history of its movements. Therefore, the newly developed features about the vessel its *speed, course, heading, angular speed* and *acceleration* are all considering the recent track history of the vessel. By adding as much information about the behaviour of the vessel to the dataset, the situation that the operator would usually observe is inserted into the dataset. This is implemented with the aim that the algorithms are able to discover patterns in the data, which comply with, and go even further than, the domain knowledge of the operator. First of all, the history of the feature *SOG* is processed into the features *speed average, speed sd, speed max* and *speed range*. These features are computed for each unique vessel. The average of the speed is computed based on all specific vessel instances of the last five minutes, unless the previous available instances cover a smaller time period. The average of the speed for a specific vessel instance is computed

FIGURE 5.14: Area types

as shown in the following equation.

$$speed_{average} = \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \tag{5.5}$$

where n is the number of instances within the last five minutes, $\overline{x}$ represents the mean speed of the considered instances and x_i represents the speed of instance i. The features that represent the standard deviation, range and maximum of the speed history are computed based on all previous available instances of the corresponding vessel track. This means the calculation for each subsequent instance considers one additional speed instance value. The standard deviation of the speed for a specific vessel instance is, therefore, computed as follows.

$$speed_{sd} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}}, \tag{5.6}$$

where n is the number of instances, $\overline{x}$ represents the mean speed of the considered instances and x_i represents the speed of instance i. The maximum of the speed for a specific vessel instance is defined as shown in the equation below.

$$speed_{max} = \max_i x_i, \tag{5.7}$$

where x_i represents the speed of instance i. The range of the speed for a specific vessel instance is defined as shown in the following equation.

$$speed_{range} = \max_i x_i - \min_i x_i, \tag{5.8}$$

where x_i represents the speed of instance i. Using the same concerning equations as for the speed calculations, the features *course average*, *course sd*, *heading average* and *heading sd* are computed. Besides these features, the angular speed and acceleration of a vessel contain information that indicates the behaviour of the vessel. The feature *acceleration* is computed for each specific vessel instance based on the current and previous speed and time of the vessel. This calculation is shown in the equation below.

$$acceleration = \frac{\Delta v}{\Delta t} = \frac{(SOG_i - SOG_{i-1}) \cdot 0.514444}{timestamp_i - timestamp_{i-1}}, \tag{5.9}$$

where the acceleration is in m/s^2, Δv is the difference in velocity in m/s, Δt is the time difference in seconds, SOG_i is the speed over ground in knots for instance i and $timestamp_i$ is the Unix timestamp of instance i. An exception to this calculation is the first instance. In this case the acceleration is set to 0, since there is no previous instance. The feature angular speed is computed for each specific vessel instance based on the current and previous COG and time of the vessel. This calculation is shown in the equation below.

$$angularspeed = \frac{\Delta\theta}{\Delta t} = \frac{(COG_i - COG_{i-1})}{timestamp_i - timestamp_{i-1}}, \tag{5.10}$$

where the angular speed is in degrees/s, $\Delta\theta$ is the difference in course in degrees, Δt is the time difference in seconds, COG_i is the course over ground in degrees for instance i and $timestamp_i$ is the Unix timestamp of instance i. An exception to this calculation is the first instance. In this case the angular speed is set to 0, since there is no previous instance. Using the same concerning equations as for the speed calculations, the features *acceleration average, acceleration sd, angular speed average* and *angular speed sd* are computed.

The addition of recent track history information about each vessel to the dataset causes the earlier instances of a specific vessel to contain less clear understanding of the situation than later instances. This, because less history can be included in the calculation in case the amount of available recent history is smaller than the determined time frame. Consequently, if historic information is important for classifying the vessel type, these earlier occurring instances have a higher probability to be wrongly predicted than later instances.

The final features that will be engineered are based on the static features *IMO, call sign* and *destination*. In general, the first two features are vessel-specific. These categorical features are, therefore, not sufficiently distinctive towards a certain vessel type. Some possible valuable information of these features can be extracted by determining whether or not the concerning feature information is available. Regarding the feature *destination*, for example, the categorical feature can be transformed into a binary feature by assigning a 1 to instances with available destination and assigning a 0 in case the specific feature value is missing. This procedure is applied to the three features, resulting in the binary features *known IMO, known call sign* and *known destination*. The same engineered feature could be made for the feature *name*, however, this feature will not result in additional information since all instances contain the corresponding value.

5.3.5 Outliers

Outliers are observations in the dataset that deviate from the other observations. In this research, it is stated that it concerns continuous feature values that are considered extreme compared to the rest of the values. Such an extreme value can be a valid data point, or can be caused by variability in a measurement or an experimental error. The detection of outliers can be performed with two different approaches. The first involves detecting univariate outliers, in which case the distribution of single feature values is examined. The other approach concerns multivariate outliers, which can be detected by examining n features in an n-dimensional feature space. These multivariate outliers are, however, intuitively difficult to interpret.

The AIS data might contain outliers, in which case should be decided how to deal with those. This depends on what the, in this specific research, classification model should be able to achieve in practice. The training dataset should represent the actual data that will be applied to the model on board of the naval vessel, as accurately as possible. Consequently, this includes the possible representation of outliers. If outliers might occur in the dataset that will be applied on board, outliers should also be present during the training process of the model.

The data analysis in Section 5.2 has shown some instances that are not included in the interquartile range of the boxplot. Some of these instances might be considered as a univariate outlier. However, it is decided to not perform outlier removal techniques and remain the possible outliers, since the nature of the training set should be similar to the real-life implementation dataset. The current dataset contains the data of message updates, which are similar to the message updates that will be received during actual implementation. Since the data might contain outliers during the actual implementation, these data points should not be removed from the training set. The model is required to adequately deal with the

outliers.

Although no outlier detection and removal techniques will be applied, one check will be performed before using the data. Very unusual observations in the data can be extracted and examined by the operator. Those observations indicate some deviant behaviour, which needs special attention and should not be left to the model to analyse. A useful continuous feature that can be used to filter those instances is the feature *SOG*. The majority of the vessels have a maximum speed of 40 knots at most. One exception is the vessel type HSC, which can reach a higher speed than 40 knots. In order to keep the high-speed crafts in the dataset, the threshold will be set 5% higher than 40 knots. Therefore, all vessels reaching a speed higher than 42 knots are removed from the dataset and left for the operator to analyse. This process will be applied to the training set, as well as the test set. Such a filter approach based on the speed cannot be implemented for each vessel type individually. This is because the exact same process cannot be applied to the test set, since the vessel type is precisely what is unavailable in the test set. Preferably, the filter is expanded for future implementation of the model, in order to detect more extreme irregularities and provide these observations to the operator to analyse. This outlier and anomaly filter is outside the scope of this research, because detecting such instances is a difficult task and suitable for a research on its own. Implementing a filter that is unrelated to the vessel type is, therefore, considered as a recommendation in Section 10.

5.3.6 Creating ground truth target labels

The debatable reliability of the AIS data has raised the desire to validate the vessel types. One validation approach is to make use of a clustering algorithm in combination with the knowledge of a human operator. The idea is to cluster the data and let the operator assign the appropriate vessel type to each cluster, based on the cluster characteristics. As an exception the whole dataset, both training and test set, will be used for this approach, since all instances need to be validated. This approach is implemented using the clustering algorithm k-means, which will be described in more detail in Section 6.1.1. The algorithm is one of the most popular clustering algorithms and is easy to interpret. It requires the input variables to be continuous, reducing the set of potential input features. The k-means algorithm is implemented for three sets of input features, which are listed below.

1. *draught, SOG*
2. *draught, SOG, COG*
3. *draught, speed max, course average, acceleration sd*

Those features are most distinctive towards the target variable based on domain knowledge and the feature importance that will be shown in Figure 5.24. The implementation is conducted for two values of k, which indicates the number of clusters the algorithm will be creating. The algorithm is implemented for the values k equal to eight and ten, since eight is the number of categories of the variable *grouped vessel type* and ten is used as the number of vessel type categories in the research of Ljunggren (2017) [19]. Before implementing the algorithm, the input features are standardised in order to scale them into the same range. Without this transformation, more weight will be on features with a higher range because of the distance-based approach. The method of standardising the features will be explained in more detail in Section 5.3.8. The outputs of the clustering algorithm do not illustrate clear and reliable clusters that can be validated by a human operator. In other words, this indicates that clustering is not an appropriate method to validate the target labels based on this dataset. This is supported by the following three issues that make validating the target labels a complex process. First, each unique vessel might contain multiple instances in the dataset that are clustered into a certain cluster, while the remaining instances of that vessel belong to another cluster. Secondly, a clustering algorithm requires the number of clusters to be defined beforehand, which makes it more difficult to experiment with the number of vessel type categories during the optimisation process. Finally, the vessels belonging to the category 'other' are likely to be spread between all categories, which makes it impossible for the k-means algorithm to cluster the 'other' vessels as one category. Moreover, the validation approach is debatable, since the knowledge of the human operator will be considered trustworthy for 100% and, therefore, considered as the truth. One of the clustering visualisations is shown in Figure 5.15. Because the clustering approach cannot contribute to a clear target validation, further analysis of the instance clusters is excluded.

A different approach to validate the vessel types of the AIS dataset is by using external databases like marinetraffic.com and digital-seas.com. These databases contain the information about the vessel type

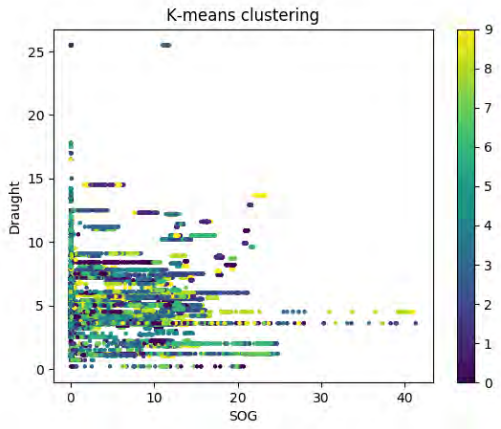

FIGURE 5.15: Visualisation of the ten created clusters based on the features *SOG*, *draught*, *COG*

and can be considered trustworthy. However, this validation approach requires too much time and is, therefore, not implemented.

Due to the complex or time-consuming nature of validating the vessel types, it is assumed that the entire AIS dataset is correct. The same reasoning as for the other categorical features in the AIS data is considered here. In general, the vessels located at the North Sea are trustworthy, which means the vessel types are rarely spoofed. Besides examining the reliability of the target feature, the imbalanced nature and the number of categories of the target feature should be considered as well. These two characteristics are undesirable, because it requires a higher complexity of the predictive model. In addition, this higher complexity might not even be achieved because of possible missing patterns between the explanatory features and the many categories of the target feature. The training process is harder when using an imbalanced dataset, since the minority classes are less represented and patterns are often hard to detect because of the lack of data. A high number of categories is undesirable, because it makes the detection of patterns hard as well. This might be caused by the lack of discriminating capability between those categories. Both characteristics can be tackled with the approach of combining several vessel types. During this approach it is important to keep note of the evenly division of the vessel type categories and the discriminating capability between and within categories. The discriminating capability should be high between categories, yet should be low within a specific category. If the differences between vessel instances within a certain category are too big, this will result in failure to detect clear patterns and, consequently, failure to predict the correct vessel type. The exact composition of the target variable also depends on what categories Thales wants to distinguish, and whether this is in accordance with the dataset. If there is a need to distinguish fishing vessels but it covers a minority of the dataset, it is debatable whether to take fishing as an individual category. The optimal number of categories is dependent on what vessel types can be combined in order to optimise the discriminating capability. Therefore, the combining process is supported by domain knowledge of the different types of vessels. The 45 vessel types are first combined into 19 categories, this conversion is shown in Table 10.4 in the Appendix. The 19 categories are the result of a logical combining approach, e.g. the two categories Cargo ship, Reserved for future use and Cargo ship, General are combined into the category Cargo ship. The number of instances in each of these combined categories is plotted in Figure 5.16. The representation of each vessel type number is displayed in Table 5.2.

A significant characteristic for the human operator to classify the vessel type is the vessel its speed. This means that all vessels anchored or moored vessels are hard to classify and are hard to distinguish. Therefore, the same histograms are created considering a subset of the dataset, in order to see whether this significantly effects the vessel type composition. Figure 5.18 and Figure 5.19 show the histogram of vessel types for the dataset without instances with SOG = 0 and for the dataset without instances containing MMSIs with mean(SOG) < 0.1, respectively. Remarkable about the first histogram is the significant decrease of the category pleasure craft and the disappearance of the vessel type reserved for future use. This indicates that a high proportion of the pleasure crafts are moored and all reserved for future use vessels are moored. The latter is self-explanatory, since vessels that are reserved for future use are unused and,

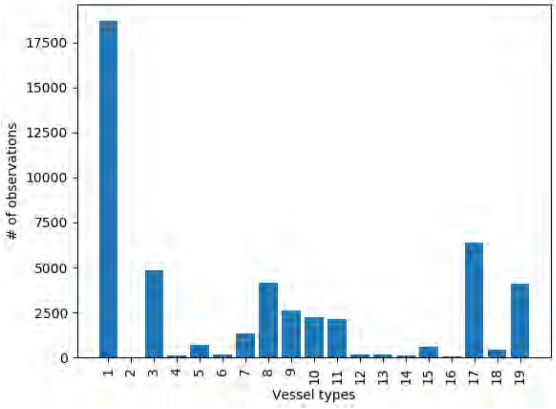

FIGURE 5.16: Histogram of combined vessel types (observations)

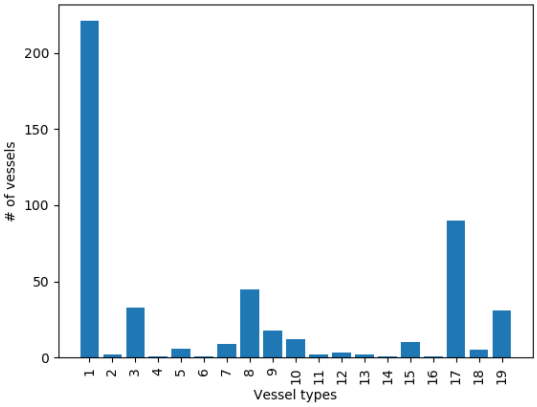

FIGURE 5.17: Histogram of combined vessel types (vessels)

therefore, have a speed equal to 0. The actual vessel type of this category is unknown, which implies that those instances should be subdivided into the other category. The second histogram is a better representation of the moving vessels, since anchored or moored vessels may produce instances containing a speed higher than 0 knots. However, the average of all instances are unlikely to be above 0.1 knots when the vessel is not sailing. The histogram in Figure 5.19 is significantly different compared to the histogram in Figure 5.18, because of the disappearance of categories engaged in diving operations and towing. This small analysis shows which categories should not be a category on their own, since the discriminating capability will be too low.

The number of categories after the first conversion is still rather high and the histograms still show extreme unevenly distributed categories. Therefore, another conversion will be implemented, preferably assigning the minority categories to the majority ones. This conversion will be supported by domain knowledge, involving three essential factors. The first consideration involves the fact that the categories must be significant for military purposes, since the end-users of the desired product are human operators on board of the naval vessels. Secondly, vessels within the same category should have similar behaviour, because this will increase the discriminating capability of the model. Finally, the conversion requires a trade-off approach between frequent categories within this specific dataset and within the general composition at sea around the world, in order to make the model generalisable. Both histograms in Figure 5.18 and Figure 5.19 show a majority of the categories cargo, engaged in dredging or underwater operations, law enforcement, other, passenger, pilot, tanker and tug. This is in accordance with the histogram in Figure 5.16, except for the additional category pleasure craft. These categories are, therefore, the basis for the next conversion. The category fishing is significant for military purposes. Moreover, the fishing category is a significant major category within the general composition at sea around the world. The specific time frame is the cause of the category minority for this specific dataset, but because of the previous two reasons this category is considered significant. The category engaged in dredging or underwater operations is not considered significant for military operations. However, this dataset covers a major part of the Dutch harbour, containing a significant amount of dredging vessels. Therefore, this category is considered as a relevant individual category. The minority category HSC is considered as individual category as well, because of its distinctive character compared to other categories. The category harbour vessel will contain the categories pilot vessel, port tender, towing and tug, since all have quite similar behaviour that is performed in the harbour area. The category sailing is combined into the category pleasure craft. The three categories engaged in military operations, law enforcement vessel and search and rescue vessel are all considered as the category military vessel. The categories cargo, tanker and passenger remain exactly as it is. Finally, all remaining vessel types are combined into the category other. This conversion of the vessel type categories is shown in Table 10.4 in the Appendix and in Table 5.2. Figure 5.22 represents the histogram of the vessel type categories after this second conversion.

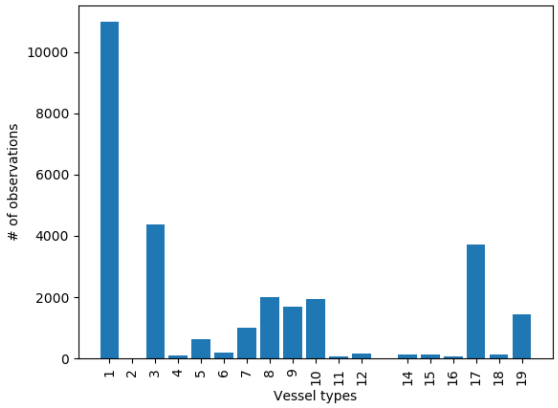

FIGURE 5.18: Histogram of combined vessel types, SOG=0 removed (observations)

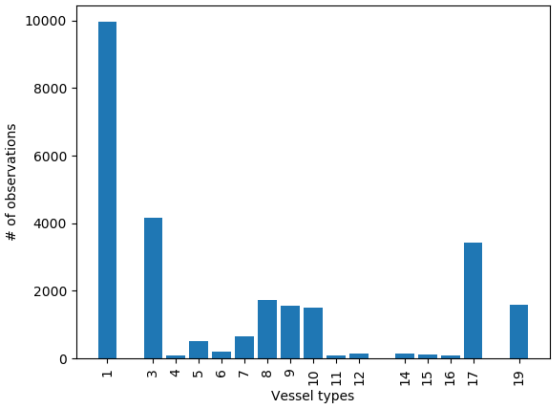

FIGURE 5.19: Histogram of combined vessel types, mean(SOG)<0.1 removed (observations)

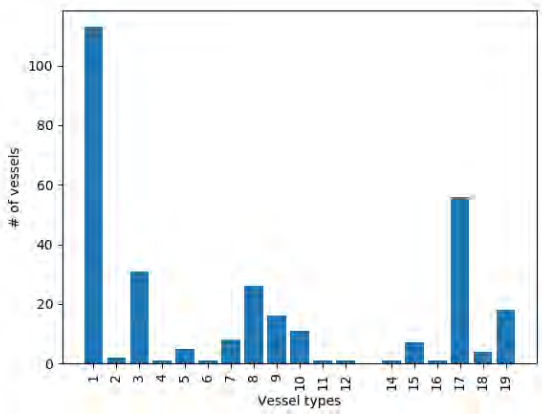

FIGURE 5.20: Histogram of combined vessel types, SOG=0 removed (vessels)

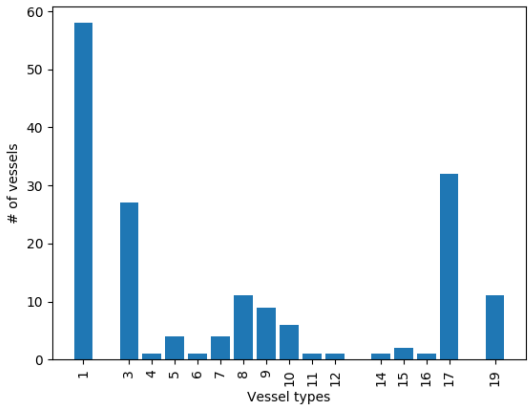

FIGURE 5.21: Histogram of combined vessel types, mean(SOG)<0.1 removed (vessels)

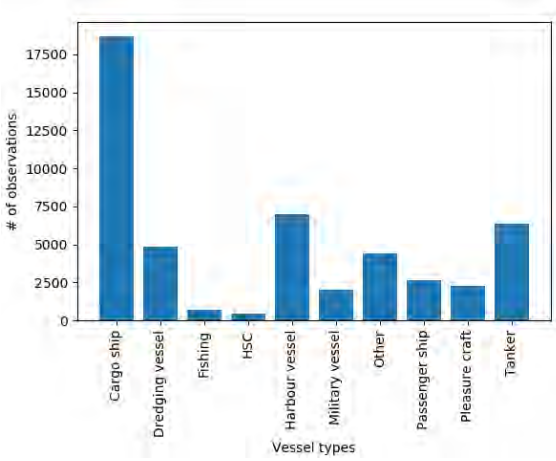

FIGURE 5.22: Histogram of initial vessel types (observations)

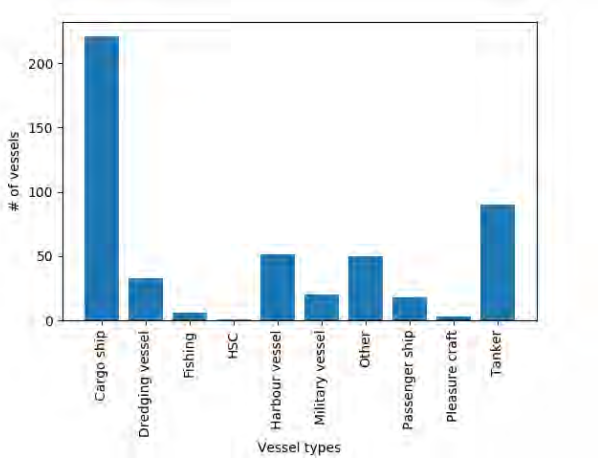

FIGURE 5.23: Histogram of initial vessel types (vessels)

TABLE 5.2: Combining the sub vessel types

Number	Combined vessel type	Combined vessel type 2
1	Cargo ship	Cargo ship
2	Engaged in diving operations	Other
3	Engaged in dredging or underwater operations	Dredging vessel
4	Engaged in military operations	Military vessel
5	Fishing	Fishing
6	HSC	HSC
7	Law enforcement vessel	Military vessel
8	Other	Other
9	Passenger ship	Passenger ship
10	Pilot vessel	Harbour vessel
11	Pleasure craft	Pleasure craft
12	Port tender	Harbour vessel
13	Reserved for future use	Other
14	Sailing	Pleasure craft
15	Search and rescue vessel	Military vessel
16	Ship not party to an armed conflict	Other
17	Tanker	Tanker
18	Towing	Harbour vessel
19	Tug	Harbour vessel

This composition of vessel types will be used as initial target variable in the iterative optimisation process. Within this process the aim is to optimise the final model performance, which will be explained in more detail in Section 6.4.1. During the iterative optimisation process can be experimented with the composition of the vessel types, in order to find a balance between a good performing model and vessel type categories that are suitable to be used in practice. Variation in composition can be applied in case discriminating capability between categories is too low. Moreover, particular care should be taken to avoid high distinctiveness within a specific category. This initial composition of the target variable will also be used as target during feature selection in the next section. However, note that a different composition could cause other features to be most relevant.

5.3.7 Feature selection

The main goal here is to select a subset of independent features, which are relevant in terms of predicting the vessel type but are not highly correlated to each other. What features have most significant predictive power may depend on the type of machine learning algorithm. Therefore, varying with the subset of input features is essential to find the best performing model. The outputs of the feature selection methods

explained below could serve as initial subset of input features and as indication of what features probably are maximally effective for building the model. Selecting a suitable subset of features is important, because of several reasons. The most obvious reason is because the dataset might contain features that do not have any added value. For example, the feature *longitude* on its own cannot add any valuable information to the model to indicate the type of vessel. Selecting this feature as input feature is, therefore, unnecessary and undesirable. Leaving this feature out will result in a lower feature dimension and, consequently, a shorter training time of the algorithm. Less amount of features also means a less complex model that is easier to interpret. Moreover, it reduces the risk of overfitting the data. Overfitting is the process of fitting the algorithm too well to the data, which results in an overly complex model. This means the model is able to identify all relevant information in the training set, but fails to make correct predictions based on new-presented data. This incapability of generalising must be avoided, since it makes the model implementation on board of a naval vessel useless. Another solution to prevent overfitting is to use a very simple classifier, because it is more robust against a high amount of features.

The first method that gives an impression about what combination of features is useful to select is calculating Pearsons and Spearmans rank correlation between the continuous variables. More specifically, a high correlation between two continuous variables implies that one of them is redundant. One of the highly correlated features should not be selected as input feature. Two highly correlated features in this research are the features *COG* and *heading*, which causes one of them to be excluded as input feature. Furthermore, the overall conclusion of the correlation analysis with all continuous features is that engineered features based on the same original feature have a high probability to be correlated with each other. This indicates the redundancy of those features except for one. An overview of the computed Pearsons and Spearmans rank correlations are shown in Figures 10.4 and 10.5 in the Appendix, respectively. How strong the relation between an explanatory variable and the target variable is cannot be determined by Pearsons correlation, since the target variable is not continuous but categorical. Hence, the following feature selection methods are performed to get a better impression of what features are redundant, highly correlated and irrelevant.

One method that is useful for determining the predictive power of the explanatory variables is to make use of the weights that are assigned to the features after training a linear support vector machine. This method can only be implemented using the linear kernel. However, it should be noted that a non-linear kernel is more appropriate for this specific classification problem. The absolute coefficient value indicates the corresponding feature importance for predicting the target categories, since it represents the feature importance for separating the target feature into the right categories. Since the support vector machine is a distance-based model, all continuous features will be standardised before applying the feature selection algorithm. A detailed description of the support vector machine algorithm, including kernels, will be given in Section 6.2.2. The output of this method using the default settings of the support vector machine involves the features *SOG, longitude, latitude, draught, speed average, speed sd, speed range, speed max, heading sd, acceleration sd, fishing, harbour, shipping* and *not defined* to be most important.

A similar approach to determine the importance of the features is by using the feature importance method included in the random forest algorithm. A detailed description of the random forest algorithm will be given in Section 6.2.1. A random forest consists of multiple decision trees, each consisting of several nodes. In each node of a decision tree the dataset is split according to a condition of a single feature. What feature is chosen as the best feature to split on is based on the Gini impurity, for classification problems. The higher the impurity decrease, the better. The Gini impurity for a variable X is computed by the sum of all impurity decrease measures of the nodes concerning a split of variable X. The calculation of the Gini impurity of feature X is shown below.

$$G(X) = \sum_{i=1}^{c} p_i \cdot (1 - p_i) = 1 - \sum_{i=1}^{c} p_i^2, \tag{5.11}$$

where c is the number of classification categories and p_i is the probability of classification i. The probability p_i is approximated by the fraction of observations labelled as category i in the corresponding set in the node. After the training process of a tree, the decrease of the weighted impurity can be calculated for each feature. This weighted impurity is computed by multiplying the Gini impurity with the probability of reaching the node, which is approximated by the proportion of instances reaching that node. The final feature importance is based on the mean decrease impurity, which is the result of averaging the weighted impurity of each feature over all trees in the forest. Those mean decrease impurity values of the

features are ranked in decreasing order, where the highest ranked features indicate the most important features. This approach has a disadvantage in case explanatory features are correlated. This might cause one of these variables to have a low mean decrease impurity, since most of its predictive power is already included in the other feature. Consequently, a particular feature might be higher ranked than another feature which predictive power is higher. The output of this method applied to this dataset is shown in Figure 5.24, which provides a clear overview of the most important features. Clearly, this approach considers the *draught* as most important feature. This feature has, however, a high percentage of missing values, which can cause the detection of patterns between *draught* and *vessel type* that might not exist in case the missing values were known.

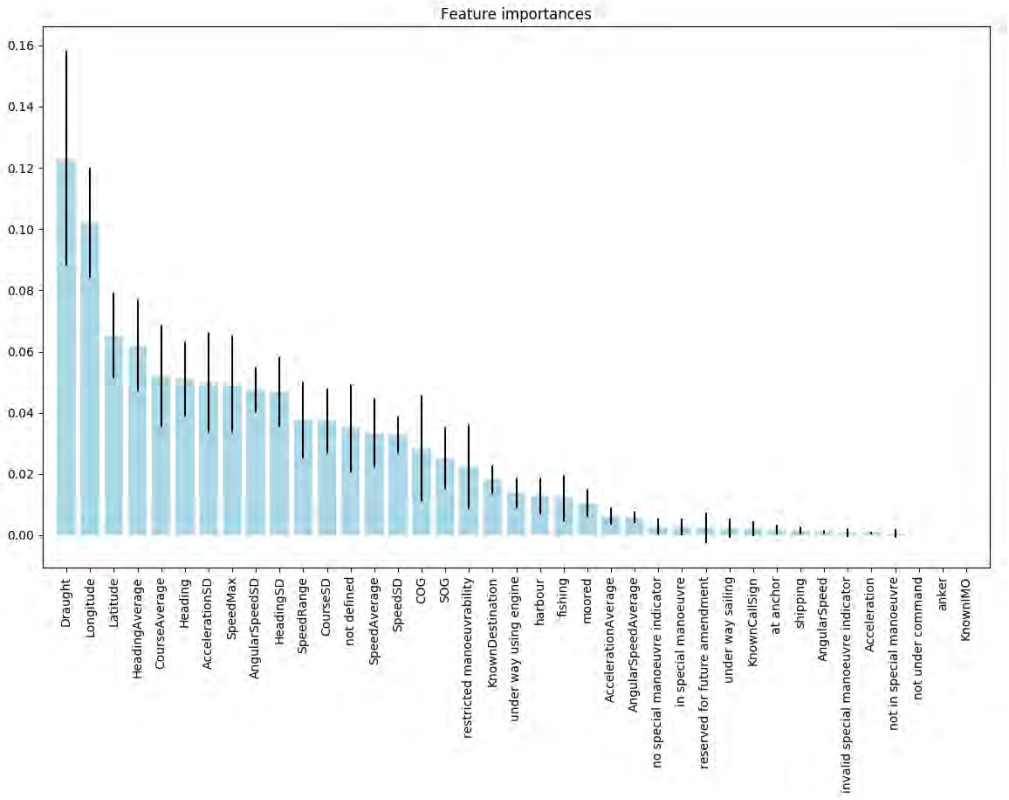

FIGURE 5.24: Feature importance based on a default random forest algorithm

An approach that results in ranked features does not necessarily indicate that the highest ranked features constitute a good subset of features. Therefore, the next considered feature selection approach is Recursive Feature Elimination (RFE). RFE is a backward feature elimination process that removes one feature in each iteration and results in a subset of features, which could be used as input features. The iterative procedure contains the following three steps.

1. Train the classification algorithm
2. Compute the feature importance for each feature
3. Remove the feature with the lowest importance

The first iteration uses all potential features in the dataset to train the classification model. Within each consecutive iteration, the classification algorithm is trained containing one feature less. During the implementation, a random forest algorithm is used as classification algorithm. This means the feature importance is based on the Gini impurity explained in the previous paragraph. The top ranked features are considered as most important features. More specifically, the last eliminated features involve the optimal final subset. However, these top ranked features are not necessarily features that are individually most relevant [11]. The stop criterion of this recursive process is reached when the subset contains as many features as the predetermined number of features. Since it is uncertain how many features are valid, this approach can be applied using cross validation, referred to as RFECV. Cross validation is a method that

tests how well a certain performance measure generalises on an independent dataset. A more detailed explanation will be provided in Section 6.4.2. Using 5-fold cross validation, the optimal number of input features will be determined for the specified machine learning algorithm in combination with the specified dataset. The different subsets of features will be scored and the subset with the highest score is selected as optimal. First, the RFE was implemented with a number of input features varying between 2 and 20. The corresponding results are included in Table 10.7 in the Appendix. The RFECV, however, showed an optimal number of input features equal to 31. The corresponding features that make this subset are included in the same Table 10.7 and are similar to the feature importance shown in Figure 5.24.

Other methods are the univariate feature selections, which covers a group of techniques that examines each feature individually. Each technique selects the best features based on some univariate statistical test. Here, the Chi-squared test is used to determine the x number of features with the highest score, which represent the features that should be selected. This test works for a classification problem like predicting the vessel type. The Chi-squared statistic is computed between each non-negative continuous normalised feature and the target feature. This test statistic for a sample of n feature observations and c mutually exclusive categories is as follows. It tests the null hypothesis whether the proportion of category i in the population is p_i.

$$\chi = \sum_{i=1}^{c} \frac{(x_i - m_i)^2}{m_i}, \tag{5.12}$$

where x_i is the number of observations of category i, $m_i = np_i$ is the expected number of observations of category i under the null hypothesis and c is the number of target categories. The 2 best features resulting from the Chi-squared test are *not defined* and *reserved for future amendment*. However, based on domain knowledge, these features are not considered most useful. This method is implemented for a number of features between 2 and 20, which results are shown in Table 10.8 in the Appendix. Note that this method is not able to examine possible highly correlated features within the selected subset, since it considers each feature individually in respect to the target feature. Therefore, this technique is not the best one to optimise the subset of features for better generalisation.

The outputs of the above-mentioned methods give an indication about the usefulness of the features. Based on domain knowledge of the operator and characteristics of the methods, a reasonable compromise of the outputs is considered as initial minimal subset of relevant features. This subset consists of the features *speed max, draught, heading average, known destination, acceleration sd, angular speed sd*, and will be the input of the first iteration in the iterative optimisation process explained in Section 6.4.1. The features *longitude* and *latitude* are excluded from most valuable feature set, since these two encourage an increase in the geographical dependency of the model. Notice that the feature selection procedures based on a random forest and support vector machine algorithm are well represented, which may give these algorithms an advantage over the neural network algorithm. This is because these selections of features are based on what optimises the predictions for that specific machine learning algorithm.

5.3.8 Feature transformation

The final process that might be necessary before using the features as input in the algorithms, is feature transformation. Whether transformation will be applied or not is algorithm specific, since it depends on the assumptions of the considered technique. Some techniques have better performance when the values of continuous variables are within a certain range, since such an algorithm is distance-based. This requires a scaling method to scale the variables values into the appropriate range. The exact scaling operations per feature will be determined based on the training set. Once the operations are defined, the same scaling is used to scale the features of the test set. This approach prevents the information to spill over between the test and training sets.

The two algorithms random forest and neural network do not require their input features to be transformed. However, a support vector machine uses a distance based approach as will be explained in Section 6.2.2, which implies the sensitivity towards big ranges in variable values. For such cases, the features should first be scaled into the best-determined range before they will be used as input variables. A potential option is to normalise the variables, which means scaling all continuous variables into a range of [0, 1]. Another possibility is to standardise the features by removing the feature mean and dividing this by the feature standard deviation. Standardising a specific feature involves applying the following equation to all corresponding instances.

$$z = \frac{X - \mu}{\sigma}, \tag{5.13}$$

where z is the standardized instance value, X is the concerning instance value, μ is the mean of the training sample of the feature and σ is the standard deviation of the training sample. After standardising, the transformed feature sample has a mean equal to 0 and a unit variance. The input features for the support vector machine are standardised before the algorithm is trained.

Another transformation that is considered during this research is transforming a categorical variable to a numerical one. Many machine learning algorithms are algebraic, which means that the input is limited to numerical variables. Particularly in this research, there is a need to transform all categorical variables into numerical, since the chosen programming packages do not contain machine learning algorithms that automatically apply this process. Since categorical input will not be accepted, these variables are transformed into dummy variables. This so-called one hot encoding is the process of transforming each category into a binary vector. Consequently, one feature column will be expanded into a number of columns equal to the number of categories. The process has significant disadvantages that should be noted. First, the dimension increases linearly with the number of variable categories, resulting in increasing training time and memory consumption. This is not an issue due to relatively less data in this research. However, if all additional features are used, the model has a greater tendency to overfit the data because an increasing number of features requires a more complex model. And a model overfits if it is more complex than another model that fits equally well [13]. In general, categorical variables have a disadvantage regardless of the one hot encoding process. Machine learning algorithms cannot deal with categories that do not occur in the training set, since no data is available to develop patterns during training the model. The categorical variables in this research are *area type, navigational status* and *special manoeuvre*. An example of the one hot encoding process is shown below for the feature *area type*. For simplicity reasons, the MMSI numbers are not in the appropriate format. This one-hot encoding method is implemented before the feature selection, since it is desirable to include the dummy variables into the feature selection algorithms.

TABLE 5.3: One hot encoding

(A) Categorical variable

MMSI	Area type
1	Harbour
2	Fishing
3	Shipping
4	Anchor

(B) Corresponding dummy variables

MMSI	Harbour	Fishing	Shipping	Anchor
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

6 Methodology

This section describes the methods that will be used to predict the vessel type of the vessels on the North Sea. First, the theoretical background of the machine learning algorithms will be discussed. This involves both the clustering method and the predictive classification methods. Then, the various methods for evaluating the model performance measures will be provided. Besides the performance measures, also the method for determining the reliability of each model performance will be discussed. Finally, the experimental setup describes how these methods will be used to optimise the performance of the predictive models, to analyse the parameter robustness of the models and to analyse the geographical dependence of the models.

6.1 Clustering method

Clustering is an unsupervised machine learning technique, which means the training data does not contain a target variable. With other words, the data is unlabelled in the sense that it is not defined into several categories. For this specific dataset it would mean that the target variable vessel type is unknown. The idea of a clustering method is to let the algorithm divide the dataset in subsets containing similar feature characteristics. Hence, this has caused the desire to cluster this data based on a clustering algorithm, assign the appropriate vessel types to the clusters based on the human operators knowledge, and validate the actual target variable with these currently assigned vessel types. The concerning implemented k-means clustering algorithm will be explained in the next section.

6.1.1 K-means

K-means is an iterative clustering algorithm that assigns each observation to one of the k clusters. Besides labelling all data observations, the final result of this algorithm consists of the k cluster centroids. The number of clusters is predetermined, since the algorithm is not able to obtain the optimal number. The data will be clustered based on specified features, which can either be all features of the dataset or only a selection of features. The algorithm starts with an initialisation of the k cluster centroids. The initial cluster centroids can either be determined by randomly selecting k observations from the dataset or by selecting k observations in a clever way to speed up the convergence. Here, the latter approach is implemented, which represents the procedure of choosing the first observation uniformly at random and the remaining k-1 observations with a high probability for observations x if it is not close any of the previously chosen centroids. The iterative procedure of the algorithm consists of two steps, which are alternated until a stopping criterion is met. In this particular implementation, the algorithm stops in case the sum of squared distances of the observations to their closest cluster center is minimized. The two iterative steps are as follows.

1. Assign each observation to the closest cluster centroid, based on the squared Euclidean distance. This means an observation x is assigned to the cluster with the following property.

$$\underset{c_i \in C}{\operatorname{argmin}}\, dist(c_i, x)^2, \quad i \in \{1, ..., k\},$$
$$dist(\mathbf{a}, \mathbf{b}) = \sqrt{\textstyle\sum_{j=1}^{n}(a_j - b_j)^2},$$

   where c_i is the centroid of cluster number i, C is the set of cluster centroids, n is the number of features considered for clustering and $dist(a, b)$ is the Euclidean distance between observations a and b. Both a and b are vectors, since the Euclidean distance will be calculated for observations containing multiple features. The set of observations that are assigned to cluster i is referred to as S_i.

2. Recompute the cluster centroids, by calculating the mean of all observations assigned to each cluster. The new centroid value for cluster number i is determined as follows.

$$c_i = \tfrac{1}{|S_i|} \textstyle\sum_{x_i \in S_i} x_i,$$

where x_i is an observation that is assigned to cluster i and $|S_i|$ is the number of observations in cluster i.

The result of the algorithm may converge to a local minimum. More specifically, the final result might not be the most optimal result, in which case it is preferential to repeat the algorithm. Another set of initial cluster centroids can lead to another optimal labelling, caused by different final cluster centroids. Therefore, repeatedly implementing the k-means algorithm is recommended.

The optimal number of clusters can be determined using several methods. However, the appropriate number of clusters might be dependent on the context. In this particular problem, the number of clusters should be consistent with the number of vessel types the operator wants to distinguish and with the vessels present in the dataset. Dividing the dataset in a certain number of clusters is especially difficult for this kind of problem, because of the other category that might contain very different type of vessels. This category is spread across the whole dataset and will, therefore, never be clustered as one.

6.2 Predictive classification methods

Predicting the vessel type is a multi-class classification problem, since the vessel type consists of more than two categories. These classifications are generated by the use of certain machine learning algorithms. This research examines which algorithms are appropriate for this specific problem and which ones suit best to this specific kind of data. Due to time limitations the three conceptual different but suitable algorithms random forest, support vector machine and neural network are implemented. These techniques are chosen because of their individual possibilities and proven usefulness in other researches. The next three subsections explain their corresponding theoretical background and implementation in Scikit-learn. All parameter names refer to the parameters of the corresponding Scikit-learn algorithm.

6.2.1 Random forest

A random forest is an ensemble method. An ensemble method combines several models in order to obtain a better performance compared to using only one of the models. The random forest as an ensemble consists of multiple decision trees. A decision tree on its own is a classification algorithm that tends to overfit. However, the random forest algorithm is more robust in this sense, because of the majority vote based on the individual decision tree predictions that determines the final prediction. The more trees in the forest, the more robust the final model is. First, the decision tree algorithm will be explained, followed by the explanation of the random forest algorithm. Both methods are explained with detail by Han et al. in the book about data mining [12].

A decision tree is a rule-based algorithm, which is designed based on the input features and the target feature. These rules can be visually represented as a tree. Such architecture is shown in Figure 6.1. The top of the tree contains the first decision node, called the root node. Each of the decision nodes corresponds to a specific feature, which will be split based on the feature value. Each feature can be used as a decision node several times in the tree. All nodes that are not split into new nodes are called leaf nodes. Each leaf node corresponds to a certain classification category, which is one of the vessel types for this specific problem. The exact implemented decision tree algorithm within Scikit-learn is an optimised version of the CART algorithm, with the exception that the algorithm only takes numerical input variables. CART stands for Classification and Regression Trees (Breiman et al. (1984)) and is an algorithm that recursively splits the dataset into two subsets based on a splitting criterion. For the classification problem, this splitting criterion is based on the Gini impurity that is shown in Equation 5.11 in Section 5.3.7, which represents the probability of obtaining two different outputs. A tree is grown using the following iterative steps for each node.

1. Find the best split for each explanatory variable.
   I.e., determine the threshold value for each explanatory variable that optimises the Gini impurity. Optimising the Gini impurity implies decreasing the impurity as much as possible.

2. Find the best split for the node.
   I.e., among the best splits determined in step 1, select the one that optimises the splitting criterion.

3. Split the node using the best split determined in step 2. Exclusively under the condition that not one of the stopping criteria is fulfilled.

The stopping criteria are as follows and indicate that the current node is not split if: [16]

- The node is pure, meaning all instances in the node have identical values for the target variable (the same classification category).
- All instances in the node have identical values for the explanatory variables.
- The tree depth reaches the maximum predetermined tree depth.
- The number of instances in a node is less than the predetermined number of instances required.
- Splitting the node will result in child nodes with less than the predetermined number of instances.
- The impurity improvement is lower than the predetermined minimum improvement for the best split.

For each leaf node, the classification category will be assigned based on the majority category of the corresponding instances.

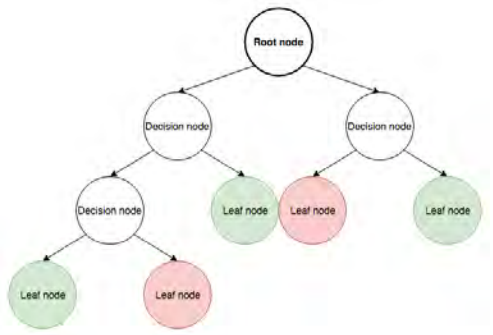

FIGURE 6.1: Architecture of a binary classification tree

Each decision tree in the random forest is built using a different subset of the data. This means the algorithm is a bagging technique, its name derived from the terms bootstrapping and aggregation. Bootstrapping is a re-sampling method that takes several subsets of the original sample. In this particular implementation the sub sample size is the same as the original sample size. Each subset is constructed by randomly sampling instances with replacement from the original sample. The models $M_1, M_2, ..., M_N$ are fitted based on the N bootstrapped samples. This is visualised in Figure 6.2 part A. For classification problems, the final prediction of the random forest model is the majority vote of all N predictions of the individual models. In other words, the results of the individual trees are aggregated, once a pre-specified number of trees are built in the ensemble. This is illustrated in Figure 6.2 part B. Besides bootstrapping, the random forest algorithm utilises another randomisation technique [21]. For each tree, the algorithm iteratively takes a random selection of the input features to build the tree. More specifically, at the root node, the best split is determined based on this random selection of the explanatory variables. All subsequent splits in the tree are based on each time another random selection of the input features. Each tree is grown until no splits can be realised. The whole decision tree building process is repeated for the predetermined number of trees.

The parameters that are important to consider during the implementation of this algorithm are shown in Table 6.1, where M equals the number of input features. During implementation, either a parameter value will be tuned using a predetermined range of values or will be set to a certain value. The maximum tree depth can be left undetermined, in which case the tree will be grown until all leaves are pure or until the number of instances in a leaf is less than min_samples_split. The min_samples_leaf parameter ensures that a split is only considered if the new generated leaf nodes contain at least this minimum number of instances after the split. Concerning the threshold for stopping the tree growth, this means the corresponding node will split if the impurity is higher than this threshold and otherwise the node will be a leaf. If the random number generator parameter is set to a certain integer value, this means the same random choices are made when the process is repeated. Consequently, this generates reproducible results. A more detailed description of the implemented random forest parameters is available on the Scikit-learn website [4].

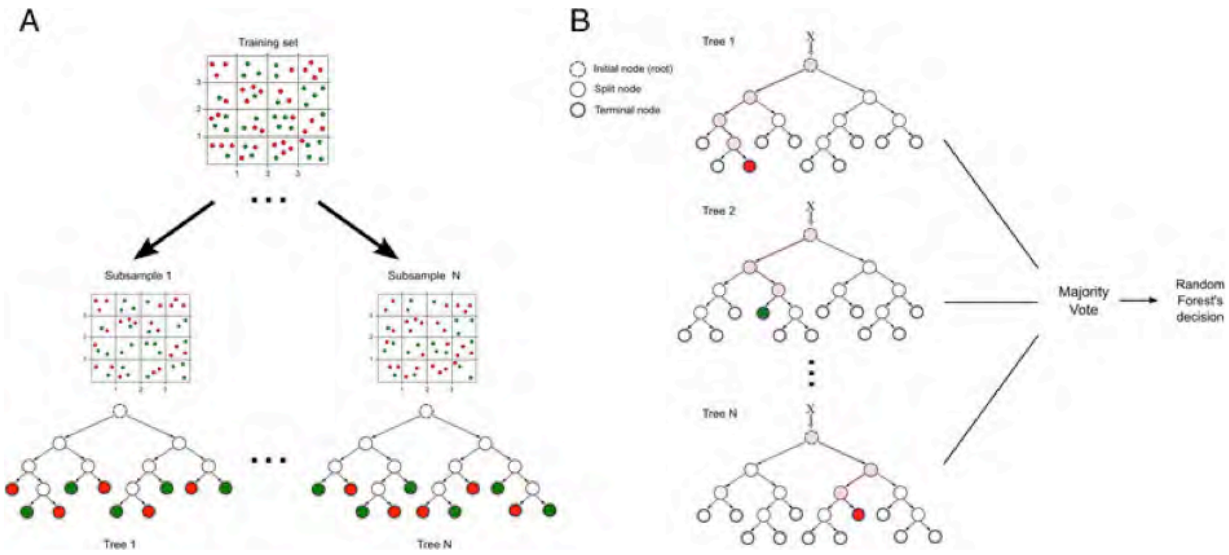

FIGURE 6.2: Visualisation of random forest [27]

TABLE 6.1: Parameters for the random forest algorithm

	Description	Range	Default	Tuned
n_estimators	Number of trees in the forest	50, 100, 200, 300, 400, 500	-	Yes
max_depth	Maximum depth of the tree	5, 8, 10, 20, None	-	Yes
min_samples_split	Minimum number of instances required in a node in order to split the node	-	2	No
min_samples_leaf	Minimum number of instances required to be in a leaf node	-	1	No
max_features	Number of features considered to determine the best split	-	$\sqrt{M}$	No
min_impurity_split	Threshold for discontinuation of tree splitting	-	1e-7	No
random_state	Determines how the random number generator is used	0	-	-

6.2.2 Support vector machine

A support vector machine is a machine learning algorithm based on splitting the data in a feature space. The input variables should be numerical, since the algorithm tries to separate the categories in the feature space. Each data observation containing M input features is represented as a vector of length M, which can be plotted in the corresponding M-dimensional input space. The algorithm objective is to find a hyperplane that is able to separate all instances of one category from all instances of the other category. Since one feature value range might be significantly different than another feature value range, support vector machines need scaling. For clear demonstration purposes, the mathematical formulation will be clarified for the binary classification problem.

Considering the training set $\{(\mathbf{x}_i, y_i)|x_i \in \mathbb{R}^M, y_i \in \{-1, 1\}, i \in 1, ..., n\}$, the aim is to find the hyperplane that ensures the largest distance between this hyperplane and the nearest data instances, called support vectors, of any category. Optimising this distance is achieved by solving the following minimisation problem [15] [6].

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i, \\ &subject\ to\ y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \end{aligned} \tag{6.1}$$

where $\mathbf{w}$ is the weight vector that is orthogonal to the hyperplane, b is the intercept term, C is the regularisation parameter, ξ_i is a slack variable and ϕ represents the function that possibly maps the input vector into a higher dimensional space. The latter will be explained in more detail in a bit. The slack variables allow the data instances to be on the wrong side of the hyperplane, i.e. to be miss-classified. It represents the distance of each data point on the wrong side to the hyperplane [7]. The regularisation parameter C allows making a trade-off between training error (the slack variable penalty) and margin maximisation (the size of the margin) [9], which consequently controls the generalisation of the classifier for unseen data. For clear visibility purposes, Figure 6.3 shows the visualisation of a support vector machine for

a linearly separable binary classification problem (number of target categories c equals 2) with the two features x_1 and x_2.

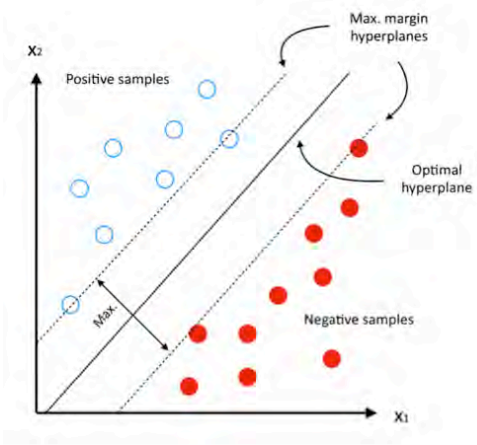

FIGURE 6.3: Visualisation of linear support vector machine

Since the support vector machine algorithm is only directly created for a binary classification problem, the multi-class classification problem needs an approach that reduces the problem to several binary problems. The used implementation in Scikit-learn uses the one-against-one approach (Knerr et al. (1990)) as default multi-class strategy [6]. This means that $(c \cdot (c-1))/2$, c representing the number of target categories, classifiers are constructed and each one will be trained based on the data from two categories. Consequently, the result of the algorithm is a set of hyperplanes.

Once a classification problem is not linearly separable because of the more complex relations between the explanatory variables, a so-called kernel function will be implemented. A kernel function scales the input vectors from dimension M into a higher dimensional space N. This implicitly mapping the training vectors into a higher dimensional space will be achieved by the function $\phi : \mathbb{R}^M \rightarrow \mathbb{R}^N$. The idea of the mapping is to be able to create the hyperplane, which is linear in dimension N, and is non-linear if the hyperplane is transformed back to dimension M. The kernel function implemented during this research is the radial basis function (RBF) kernel, since this is in general a reasonable first choice [15] [29]. The radial basis function kernel is as follows.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2}, \gamma > 0. \tag{6.2}$$

The parameters that are important to consider during the implementation of this algorithm are shown in Table 6.2, where rbf stands for radial basis function and ovo represents the one-against-one approach. In order to find the best combination of C and gamma, it is found to be most beneficial to try exponentially growing sequences of the parameter values during a grid search [15]. A more detailed description of the implemented support vector machine parameters is available on the Scikit-learn website [5].

TABLE 6.2: Parameters for the support vector machine algorithm

	Description	Range	Default	Tuned
C	Penalty parameter of the error term	10, 100, 1000, 10000, 100000	-	Yes
kernel	The specific kernel that will be used in the algorithm	-	rbf	No
gamma	Kernel coefficient	1e-9, 1e-7, 1e-5, 1e-3, 1e-1	-	Yes
decision_function_shape	Determines whether a classifier is built for each 2-pair category combination or whether a classifier is built for each category fitted against all other categories	-	ovo	No
random_state	Determines how the random number generator is used	0	-	-

6.2.3 Artificial neural network

An artificial neural network is a complex mathematical model that is inspired by the performance characteristics of the biological neuron networks. The neurons represent the position in the network where the information processing is carried out. The connections between the neurons transport the signals from one neuron to the next connected neuron, each of which has an associated weight. The signal is multiplied with the corresponding weight and this result serves as the input for the next neuron. In each neuron, the input value is transformed into the output value using an activation function, which will be

explained in more detail later this section.

The implemented type is the multi-layer perceptron (MLP) classifier, which is a feed forward artificial neural network that trains the data using backpropagation. Backpropagation will be clarified later in this section. A neural network is called a feed forward neural network if the connections between neurons exclusively point forward, which means the network does not contain any cycles. The information is fed forward from one layer to the next. Figure 6.4 is showing the architecture of an MLP classifier containing one hidden layer. The number of neurons in the input layer equals the number of input variables. The number of neurons in the output layer equals the number of output variables, which is equal to the number of categories if the target variable is multi-categorical. The number of hidden layers in a network is proportional to the complexity of the model. For most networks one hidden layer provides a sufficiently complex model and the number of neurons is generally between the amount of input neurons and the amount of output neurons [23].

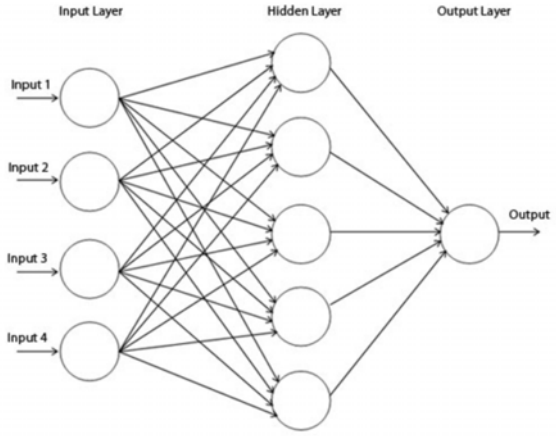

FIGURE 6.4: Visualisation of an artificial neural network

The process of the signal transformation in a neuron can be visualised as shown in Figure 6.5. For neuron k the following calculation holds to determine the corresponding output.

$$y_k = \varphi\left(\sum_{i=1}^{m} w_{ki} \cdot x_i + b_k\right), \tag{6.3}$$

where y_k indicates the output of neuron k, $\varphi(.)$ is the activation function, w_{ki} indicates the weight belonging to input i of neuron k, x_i indicates the value of input i and b_k is the possible bias term for neuron k. Applying this formula for each of the neurons will calculate the output values of the network based on the input values.

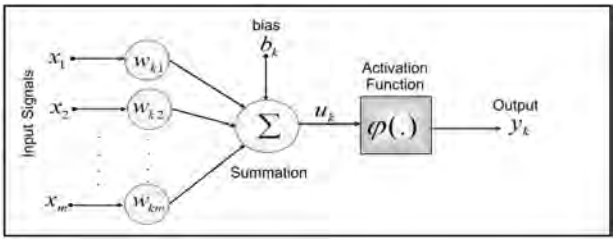

FIGURE 6.5: Structure of an artificial neuron k in an artificial neural network [26]

Scikit-learn provides several implementation possibilities regarding the activation function. In this research, the influence on the performance values of the following three most popular activation functions

are examined, which are the logistic, the hyperbolic tangent and the rectified linear units activation functions, respectively.

$$\varphi(x) = \frac{1}{1 + e^{-x}}, \tag{6.4}$$

$$\varphi(x) = \tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{6.5}$$

$$\varphi(x) = \max(0, x). \tag{6.6}$$

The objective of this algorithm is to determine the best-suited weights in order to achieve the best predictions. The predictions are determined optimal if a minimum error between the output value and actual target value is achieved. The algorithm initialises the weights randomly and subsequently optimises the weights using backpropagation. Backpropagation is an iterative updating process of the weights, which makes use of the weight updating formula shown in equation 6.8. The process has its name, backward propagation of errors, due to the fact it optimises the error function E by starting the weight updating at the final layer and moves it way backward in the network. The error function E will be calculated by the sum of the squared difference between the actual target value and the output value of the network.

$$E = \left(\sum_{k \in K} (o_k - t_k)^2 \right), \tag{6.7}$$

where k is the number of target categories, o_k represents the output value of output neuron k and t_k is the actual target value of output neuron k. Subsequently, the weights can be updated using an equation of the following form.

$$w_i \leftarrow w_i - \alpha \cdot \frac{\partial E}{\partial w_i} - \alpha \cdot \lambda \cdot w_i, \tag{6.8}$$

where w_i is the value of weight i, α is the learning rate, E is the error function and λ is the weight decay parameter. The learning rate controls the step size during updating the weights. The weight decay parameter is the regularisation term and helps prevent overfitting. The algorithm continues until the number of maximum iterations has been reached or the value of the error function is sufficiently small, which means the value is not improving by at least a predetermined number for a predetermined number of consecutive iterations. The exact solver method used during the implementation is the stochastic gradient-descent optimisation process. For more details, it is advised to consult the work of Kingma and Ba [17]. The exact implementation of the algorithm within Scikit-learn is more complex and can be found on the website [3].

For multi-class classification problems, the algorithm requires the output layer to have the number of neurons equal to the number of target variable categories. Just before the output layer, the algorithm applies the softmax function. This softmax function involves scaling the vector with real values into the corresponding vector having all values within the range [0, 1] and the sum of those values equal to 1. In other words, the softmax function assigns probabilities to each category of the multi-class classification. The output is the target category with the highest probability. The concerning softmax function is as follows.

$$softmax(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}}, i = 1, ..., c, \tag{6.9}$$

where z_i represents the ith element of the input to the softmax (corresponding to category i) and c is the number of target categories in the multi-class classification problem.

The parameters that are important to consider during the implementation of this algorithm are shown in Table 6.3. The alpha parameter controls the magnitude of the penalty for complex models. A more

detailed description of the implemented multi-layer perceptron classifier parameters is available on the Scikit-learn site [2].

TABLE 6.3: Parameters for the artificial neural network algorithm

	Description	Range	Default	Tuned
hidden_layer_sizes	The ith element indicates the number of neurons in the ith hidden layer	4, 6, 7, 8, 9, 10, 11	-	Yes
activation	Activation function	logistic, tanh, relu	-	Yes
alpha	L2 penalty parameter	1e-5, 1e-3, 1e-1	-	Yes
learning_rate_init	The starting value for the learning rate	-	0.001	No
max_iter	Maximum number of iterations	100, 500	-	Yes
random_state	Determines how the random number generator is used	0	-	-

6.3 Evaluation

Several performance measures exist to evaluate the outcome of model predictions. Different problems ask for different specific evaluation measures to be optimised in order to obtain the desired outcome. Which evaluation measures are best to optimise depends on the characteristics of the dataset and on what the prediction model should accomplish. In this particular problem, the vessel types need to be accurately predicted. The two considered evaluation measures are the classification accuracy and the macro-average F1 score. The accuracy is suitable for evaluating this specific problem, because it indicates the general performance of how many instances have been predicted correctly. The macro-average F1 score, however, is a weighted average of the precision and recall, explained in Sections 6.3.3 and 6.3.4, and gives equal importance to each class. This gives more insight in how good the predictions are for each class. All metrics are based on the confusion matrix. Therefore, this metric is briefly explained first.

6.3.1 Confusion matrix

The confusion matrix is a metric for summarising the performance of a classification model. This research involves a multi-class classification problem, which corresponds to a confusion matrix with multiple classes. However, because of readability and interpretation of determining the precision, recall and F1 score, the binary confusion matrix is explained here. In the binary case the target variable, the variable that will be predicted, consists of two classes. Each target class can be predicted correctly or incorrectly, the latter indicating the other class is predicted. The number of correct and incorrect predicted instances can be displayed in a confusion matrix, shown below in Figure 6.6.

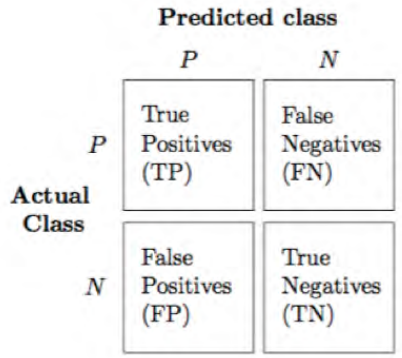

FIGURE 6.6: Confusion matrix binary classification problem

To elaborate the confusion matrix, take as an example the two target classes for vessel type as cargo and not cargo. The confusion matrix distinguishes the following four different kinds of predictions.

- True Positive (TP): an instance that is predicted as Positive and actually is Positive
  e.g.: the vessel type is classified as cargo and actually is cargo
- False Positive (FP): an instance that is predicted as Positive but actually is Negative
  e.g.: the vessel type is classified as cargo but actually is not cargo
- False Negative (FN): an instance that is predicted as Negative but actually is Positive
  e.g.: the vessel type is classified as not cargo but actually is cargo

- True Negative (TN): an instance that is predicted as Negative and actually is Negative
  e.g.: the vessel type is classified as not cargo and actually is not cargo

Transforming this into a multi-class classification problem, the confusion matrix expands in number of actual classes and number of predicted classes to the number of target classes. In which case the concept of TP, FP, FN and TN is no longer appropriate. However, for the multi-class problem those values could be defined using a one-against-all approach, which calculates the values relatively to each class. This entails the following definitions for each class c.

- TP(c): an instance that is classified as c and actually is c
- FP(c): an instance that is classified as c but actually is not c
- FN(c): an instance that is not classified as c but actually is c
- TN(c): an instance that is not classified as c and actually is not c

6.3.2 Accuracy

Intuitively the first performance measure that comes to mind is the classification accuracy. It is an overall performance measure that calculates the percentage of correctly predicted instances. The corresponding formula for a binary classification problem is as follows.

$$Accuracy = \frac{Number\ of\ correctly\ predicted\ instances}{Total\ number\ of\ instances} = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.10}$$

In general, the accuracy value is calculated by dividing the sum of the diagonal numbers in the confusion matrix by the sum of all numbers in the matrix. I.e., the denominator of the accuracy equation is equal to the sum of all values within the confusion matrix, and the numerator is equal to the sum of TP's of each class.

Considering the classification accuracy as a performance measure can be misleading. More specifically, the accuracy withholds interesting information about the errors in case the classification target has more than two classes or in case the target classes are not evenly distributed. An obvious disadvantage is that a high accuracy can be achieved with a weak classifier that always predicts the most frequent class. Another example, if the least occurring class is correctly predicted for every instance, this has little effect on the overall accuracy outcome. This kind of information can be derived from the confusion matrix. However, the confusion matrix is not a one-value evaluation measure, which is preferable and needed for optimising a machine learning algorithm. Therefore, the macro-average F1 score is considered as evaluation measure in addition to the classification accuracy.

6.3.3 Precision

The precision is a metric that indicates the proportion of positive classified instances that are actually positive. For a multiclass classification problem this performance measure can be determined using the one-against-all approach. The precision for class c is expressed in the following equation.

$$Precision(c) = \frac{TP(c)}{TP(c) + FP(c)} \in [0, 1] \tag{6.11}$$

Alternatively, this value can also be determined using the original confusion matrix X including all classes. That equation is as follows.

$$Precision(c) = \frac{X_{cc}}{\sum_i X_{ic}} \in [0, 1], \tag{6.12}$$

where $X_{i,j}$ represents the number in row i and column j in the confusion matrix.

6.3.4 Recall

The recall represents the proportion of actual positive instances that are classified as positive. As with calculating the precision, the recall for a multi-class classification problem can be determined using the one-against-all approach. The recall for class c is expressed in the following equation.

$$Recall(c) = \frac{TP(c)}{TP(c) + FN(c)} \in [0, 1] \tag{6.13}$$

Alternatively, this value can also be determined using the original confusion matrix X including all classes. That equation is as follows.

$$Recall(c) = \frac{X_{cc}}{\sum_i X_{ci}} \in [0, 1], \tag{6.14}$$

where $X_{i,j}$ represents the number in row i and column j in the confusion matrix.

6.3.5 Macro-average F1 score

The F1 score is an evaluation measure that combines the harmonic mean of the precision and recall into one performance value. This measure can be calculated for each target class c individually, which is shown in the following equation.

$$F1\ score(c) = 2 \cdot \frac{precision(c) \cdot recall(c)}{precision(c) + recall(c)} \in [0, 1] \tag{6.15}$$

Since an overall evaluation value is preferred over a value for each target class individually, the F1 scores are combined into one value. This combination of values can be accomplished using several approaches. One approach that considers the class imbalance is the macro-average approach, as it gives equal importance to each class. This results in the final macro-average F1 score, which is calculated as follows.

$$Macro - average\ F1\ score = \frac{\sum_{c \in C} F1\ score(c)}{n} \in [0, 1], \tag{6.16}$$

where C is the total set of target classes and n is equal to the size of set C (the number of distinctive categories). This performance measure has a disadvantage if a situation arises where one of the target categories does not exist in the test set. This means the specific category has no appearing true positives and, therefore, always will ensure the precision and recall to be equal to 0. As a direct consequence, the corresponding F1 score of this category is equal to 0. Since the macro-average gives each class equal importance, this value of 0 has a big influence on the overall macro-average F1 score.

Since both the accuracy and the macro-average F1 score have their disadvantages, both performance measures are reflected during evaluation.

6.3.6 Confidence interval

The reliability of the model performance is evaluated using a bootstrap procedure, which is a method appropriate to create the desired confidence intervals. Such a confidence interval gives an indication of what the performance measure will be between in 95% of the cases. In other words, the interval estimate is 95% certain to contain the true value of the unknown sample model performance measure. The bootstrapped confidence interval will be determined using the following steps.

1. Randomly draw with replacement B = 1000 samples of the original dataset with the same size.
2. For each of the B bootstrap samples, randomly select 80% of the unique vessels and consider these as the training dataset. The remaining unique vessels are considered as test set. Each of the training datasets will be used to train the selected machine learning algorithm. Each obtained model is evaluated using the corresponding test set by calculating the selected performance measure. These B performance measures are considered as the empirical distribution, $\theta_1^*, ..., \theta_B^*$.
3. The empirical distribution will be used to determine the 95% bootstrap confidence interval, which is constructed by taking the 2.5% and 97.5% percentiles from the B performance measure values. This results in the bootstrap confidence interval $[\theta^*_{(\alpha/2)}, \theta^*_{(1-\alpha/2)}]$, where $\theta^*_{(\alpha/2)}$ denotes the $\alpha/2$ percentile of the bootstrapped coefficients θ^* and α is equal to 0.05.

A small 95% confidence interval is preferred, since this indicates a reliable model. Alternatively, a wide confidence interval represents an unreliable model that could perform significantly worse or better if another dataset is used.

6.4 Experimental setup

This section discusses exactly how the experiments are conducted in order to answer the research questions. It involves the optimisation and evaluation of each model performance, utilising the previous mentioned methods of Section 6.2 and 6.3. First, the steps of the iterative optimisation process that is common within machine learning will be explained. The results of this process are the final model characteristics. Subsequently, the other subsections present the design of the parameter robustness analysis, the implementation of model evaluation and the approach to test the geographical dependence.

6.4.1 Iterative optimisation process

The common machine learning process of optimising a model performance is iterative, allowing each iteration to change aspects of the model. In each iteration specific features are selected, parameter values are chosen and a different composition of target categories may be possible. A specific selection of these model characteristics results in a particular performance outcome. Based on the effect of the changes in model characteristics on the performance will be determined what changes are implemented in the next iteration. The goal of each consecutive iteration is to improve the previous performance, with the ultimate aim to reach the optimum. In case no increasing performance is observed in several successive iterations, the iterative optimisation process is ended. This does not necessarily indicate that those model characteristics lead to an optimum, much less the global optimum.

More specifically, the possible changes within each iteration related to the input features, parameter values and composition of the target feature are not based on specific predetermined rules, but are rather based on exploratory testing. Using a trial and error approach the accuracy as performance measure is to be optimised. A good model performance will be achieved when the model has high predictive power. A high predictive power for this specific problem is generated by a high discriminating capability between the target categories. This will be evaluated by adequate performance measures. The initial and main focus is optimising the accuracy, because of the moderate initial performance. Once the model performance increases, the macro-average F1 score increases in importance simultaneously. For each of the three concerning classification algorithms, each iteration implements at least one of the following steps 1, 2 and 3, and always ends with step 4.

1. Select features
   The exact input features should be determined. The composition and the amount of features might be different in each iteration. Varying this gives an indication of what features and what amount of features lead to the best results for that specific algorithm. Since each algorithm has a different approach to the actual feature values, scaling continuous data may be essential for the performance of the model. The appropriate scaling method is established based on common knowledge about each algorithm. The selected features in the first iteration will be determined by the output of the feature selection algorithms and the correlation analysis, both discussed in Section 5.3.7.

2. Choose parameter values
   The exact parameter settings can have significant impact on the performance. Therefore, finding the appropriate parameter values is essential and will be achieved by tuning the specific parameters of every machine learning technique. These values should be specified in advance, because they cannot be learned by the algorithm itself. The concerning parameters of each implemented algorithm are explained in Section 6.2, as well as the parameters that will be tuned for each algorithm. The parameter values will be determined based on most common values for similar models in the literature.

3. Determine composition of target feature
   The target feature of this classification problem is the vessel type. However, as is learned from the data analysis, many types exist that can be combined into the same vessel type. Varying the composition can help determine how the vessels should be divided into categories to get the highest discriminating capability. This means the number of categories and the composition of the categories should be experimented. The changes in composition should be determined based on domain knowledge, to ensure a rational combination of vessel types. The composition of the target categories in the first iteration is equal to the one shown in Section 5.3.6 Figure 5.22, which is a balance between maintaining as much visible distinctive vessel categories and reducing the number of categories.

4. Analyse performance
   After establishing previous three steps, the model can be trained and evaluated. During the training process the algorithm tries to find patterns between the input variables and the target variable. Subsequently, the same explanatory variables of the test set will be used as input in the trained model in order to make the predictions. These predictions will be evaluated by comparing it to the actual target values of the test set. The effect of the adjustments on the performance measures will be analysed. It is compared to the performance of the previous iteration and is analysed what would provide an even better performance. Those interpretations will be implemented in the next iteration.

Such an iterative optimisation process continues until no significant changes are observed. The final specific experimental conditions create the experimental results for each of the three algorithms. With regard to the range of parameter values, these will be substantiated in the next section. The various combinations of selected features and different compositions of the target feature, however, are not included in this thesis due to clear representation purposes. The final selected features and the composition of the target feature used in each trained model will be stated in the Section 7 Experimental results and evaluation.

6.4.2 Parameter robustness analysis

A method to check the robustness of the models is a parameter robustness analysis. The analysis follows a certain approach that is based on varying the optimal determined parameter values, in order to obtain the effect on the performance. The iterative process to determine the optimal parameter values is implemented with the grid search cross validation approach within Scikit-learn. This method also satisfies the approach to investigate the parameter sensitivity, since it involves the model performance results originating from different combinations of parameter values.

Cross validation is a model validation method to check how the model performance generalises to an independent dataset. This method provides a more general and reliable impression about the effect of the specific model characteristics, since it considers multiple independent training and validation sets to evaluate the performance. Here, 5-fold cross validation will be applied to prevent overfitting, which means the training dataset determined in Section 5.3.2 is randomly split into five approximately equal folds. Each fold is then used as a validation set while the four remaining folds together are used as training set. Since the main performance measure is the accuracy, this measure is used to estimate the performance of the model tested on the validation set. The output of 5-fold cross validation equals five times an accuracy, which is averaged to obtain the final estimate. Each combination of training and validation set should not contain the same unique vessel. Since the existing Python code does not take into account the division of unique vessels in either the training or validation set, the trained model might be biased. Important to keep in mind is that this can result in an estimated performance measure higher than the actual performance measure value. Thus, the final performance measure tested on the test set might be lower.

Grid search is implemented to optimise the parameter values. This approach tries all combinations of the manually specified set of parameter values, which are commonly used default values, in order to find the best performing combination. The exact ranges of the implemented parameter values for each of the three machine learning algorithms are shown in Tables 6.1, 6.2 and 6.3. Concerning the reproducibility of the results, the random_state parameter is set to a seed of 0 for all algorithms. Consequently, the same splits of training and validation folds will be made when the procedure is repeated.

- For the random forest algorithm the random_state parameter also ensures similar selection of features to determine the best split. The parameter n_estimators is varied from 50 to 500 trees in order to find the balance between model improvement and computation time. The max_depth of the tree is implemented for a minimal depth of 5 and a maximum depth of undetermined. For all remaining parameters the default value is used.
- The kernel used for all implemented support vector machines is the RBF kernel. This because it is the most common used kernel in case a problem is not linearly separable. For this particular kernel it is recommended to implement exponentially growing sequences of the parameter values of C and gamma.
- The number of hidden layers in the artificial neural network is set to one. Zero hidden layers is not desired, since the model is not linearly separable. However, more than one hidden layer is

computational expensive, since it increases the model its complexity level. Therefore, the number of hidden layers is set to one. The number of neurons in the hidden layer is varied from 4 to 11, since most of those are between the number of neurons in the input layer and output layer. For certainty reasons, 4 and 11 number of neurons are considered as well. The activation parameter is implemented with the logistic, the hyperbolic tangent and the rectified linear units activation function. The first two are the most common ones, as the latter does not suffer from the vanishing gradient. The parameter alpha is varied with three commonly used values 1e-5, 1e-3 and 1e-1. The maximum number of iterations, max_iter, is implemented with a maximum of 100 and 500. This tests whether the model performance is significantly different when the algorithm has not converged (max_iter = 100) compared to when the algorithm does converge (max_iter = 500). The default value for the initialisation of the learning rate, learning_rate_init, will be used.

The output of the grid search cross validation contains the mean performance measure and the standard deviation for each of the considered combinations of parameter values. This immediately provides an overview of the robustness of the model performance concerning the parameters. Based on the output of this approach, the best combination of parameter values for each of the algorithms can be determined.

6.4.3 Model evaluation

After determining the best combination of parameter values for each of the three algorithms, these values will be used to train the models based on the whole training dataset and test them based on the same test set for a fair comparison of the three models. To validate the reliability of the models, the bootstrap procedure explained in Section 6.3.6 will be implemented to construct a confidence interval for each machine learning algorithm and each performance measure. This results in six confidence intervals, based on the whole dataset.

6.4.4 Geographical dependence analysis

Another method to check the robustness of the models is to apply the same model to datasets originating from geographical different locations. The possible effect of the geographical dependence follows from comparing the performance. This effect will be evaluated using the model properties resulting from the iterative optimisation process, which are optimised based on the standard dataset used for this research. The predictive power robustness of these model properties are tested based on three different situations, for each of the three considered machine learning techniques.

1. Using the model as it is after applying the iterative optimisation process on the standard train set. The model performance is tested based on the standard test set originating from Section 5.3.2.

2. Using the model as it is after applying the iterative optimisation process on the standard train set. The model performance is tested based on a test set originating from a different geographical dataset. This dataset originates from Den Helder, which also covers part of the North Sea but is predominantly close to Den Helder itself.

3. Using the model properties as it is after applying the iterative optimisation process on the standard train set. However, these model properties will be used to train a model based on a train set originating from the Den Helder dataset. Important is to keep the size of this train dataset equal to the size of the standard train set. The performance of the resulting model is tested based on the test set originating from the Den Helder dataset (the same test set as in situation 2).

Once the performance of at least situation 3 is significantly worse than the performance of situation 1, this indicates a geographical dependence. In case there is no significant difference between situation 1 and situation 2 or 3, there is no evidence for geographical dependence. However, the two concerning datasets have some interdependence, since both cover part of the North Sea. This means no firm conclusion can be made towards geographical dependence if the difference in performance is small. Geographical dependence can certainly matter if the absolute distance between the geographical location of the datasets is much bigger. In case geographical dependence is observed, this indicates that the iterative optimisation process should be applied to a new dataset with a geographical different location before the model can be used to predict the vessel type.

7 Experimental results and evaluation

This section presents the results of this research. In the iterative optimisation process the performance is optimised by varying the selected features, the parameter values and the composition of the target variable. This thesis only contains the performance results regarding varying parameter values to keep focus and not make the thesis too long. The final composition of the target variable *vessel type*, as explained in Section 5.3.6, and the final selected features for each of the three algorithms, as identified and justified in the corresponding sections below, are established as fixed. First, the optimal parameter values of each model as found after implementing the grid search approach will be provided. Then, these optimal parameter values will be used to train the models on the training set and evaluate each model performance on the independent test set. Finally, the reliability of each model performance is evaluated by the constructed bootstrap confidence interval. This section ends with a comparison between the three optimised models. The approach to test the geographical dependence of the models has not been implemented due to lack of time.

7.1 Random forest

The final selected features for training the random forest algorithm are based on the output of the RFECV approach, explained in Section 5.3.7, and the feature importance shown in Figure 5.24. The selected features are the highest ranked features in Figure 5.24 up to and including the feature *moored*, except for the features *longitude* and *latitude*. This concerns the following selection.

- *SOG*
- *COG*
- *heading*
- *draught*
- *speed average*
- *speed sd*
- *speed range*
- *speed max*
- *course average*
- *course sd*
- *heading average*
- *heading sd*
- *known destination*
- *acceleration sd*
- *angular speed sd*
- *fishing*
- *harbour*
- *moored*
- *not defined*
- *restricted maneuverability*
- *under way using engine*

The optimal parameter values resulting from the grid search approach are the same for both optimising the accuracy and the F1 score, which are as stated in Table 7.1. The cross validation performance values are 0.592 (+/-0.078) and 0.604 (+/-0.058) for the accuracy and F1 score, respectively.

TABLE 7.1: Optimal tuned parameter values for each optimised performance measure (random forest)

	Accuracy	**F1 score**
n_estimators	500	20
max_depth	500	20
Cross validation performance	0.592 (+/-0.078)	0.604 (+/-0.058)

The cross validation results for each parameter set during the grid search are shown in Table 10.9 in the Appendix for optimising the accuracy and F1 score. Considering the accuracy as performance measure, the overall results show no significant difference between the performance values of each considered parameter set. Especially varying the number of trees in the forest shows robust results in case all other parameters are fixed. However, the maximum depth of the tree has more impact, since the trend of the performance value is proportional to the maximum depth of the tree. In general, the standard deviation of the performance measure also seems to have a positive relation with the maximum depth of the tree. The biggest difference between the average model accuracies for this exact implementation is 5.2%. With regard to the F1 score, the different parameter sets have more influence on the performance. More specifically, the biggest difference between the average model F1 scores for this exact implementation is

equal to 14.1%. The standard deviation, remarkably, does not generally increase with an increasing maximum depth of the trees like it does for the accuracy. The best performing parameter values are used to train the random forest and, subsequently, test the model performance on the independent test set. The corresponding performance involves a classification report and confusion matrix, which can be found in Tables 7.4 and 10.10 in the Appendix, respectively. The confusion matrices of all three obtained models are analysed in Section 7.4 Model comparison. The final accuracy and F1 score are as follows.

Accuracy = 0.4960 F1 score = 0.2477

The accuracy value of nearly 0.5 does not mean that a coin can be flipped and achieve a similar performance. Since the target variable contains multiple unevenly distributed categories, obtaining a high accuracy is harder compared to two evenly distributed categories.

Using the approach, explained in Section 6.3.6, the bootstrap confidence intervals for both the accuracy and F1 score are constructed. As shown below, it can be concluded that the accuracy of a trained random forest model based on this dataset is with 95% certainty between 0.3622 and 0.5903. With regard to the F1 score, the performance value is with 95% certainty between 0.1736 and 0.3634. These rather big ranges indicate that the performance of the model is strongly dependent on the data. This, because a different combination of training and test set can provide significantly different performance values. This means the reliability of the final random forest model is dubious.

Accuracy = [0.3622, 0.5903] F1 score = [0.1736, 0.3634]

7.2 Support vector machine

The final selected features for the support vector machine algorithm are based on the most important features for a linear support vector machine with a penalty parameter C equal to 1000, which feature selection method is explained in Section 5.3.7. This set of features seems to remain the most important features for a support vector machine with a varying value of the penalty parameter. Again, the features *longitude* and *latitude* are removed from this set of features. The specified features are as follows.

- *SOG*
- *draught*
- *speed sd*
- *speed range*
- *heading sd*
- *acceleration sd*
- *fishing*
- *harbour*
- *shipping*
- *not defined*

The optimal parameter values resulting from the grid search approach are shown in Table 7.2, for both optimising the accuracy and the F1 score. The cross validation performance values corresponding to these optimal parameter values are 0.514 (+/-0.084) and 0.503 (+/-0.061) for optimising the accuracy and F1 score, respectively.

TABLE 7.2: Optimal tuned parameter values for each optimised performance measure (support vector machine)

	Accuracy	**F1 score**
C	100	1000
gamma	1e-3	1e-1
Cross validation performance	0.514 (+/-0.084)	0.503 (+/-0.061)

The cross validation results for each parameter set during the grid search are shown in Table 10.11 in the Appendix for optimising the accuracy and F1 score. The exponentially increasing value of the parameter C leads to increasing accuracy results for gamma values equal to 1e-7 and 1e-5. For the three other gamma values, a clear relationship is less obvious. This indicates that the penalty parameter does have considerable influence towards better performance, though with the right combination of the gamma value. The gamma parameter itself has most optimal accuracy values occurring for gamma values equal to 1e-3 or 1e-5. And, especially the models with a gamma value equal to 1e-9 and 1e-1 perform significantly worse. Overall the biggest difference in mean accuracy between the different parameter sets is

equal to 13.5%. The best performing parameter values regarding optimising the accuracy are used to train the support vector machine and, subsequently, test the model performance on the independent test set. The corresponding performance involves a classification report and confusion matrix, which can be found in Tables 7.5 and 10.12 in the Appendix, respectively. The final accuracy and F1 score based on the test set are as follows.

Accuracy = 0.5636 F1 score = 0.2957

The reliability of these performance values is assessed with the corresponding bootstrap confidence intervals, which are shown below. The accuracy is with 95% certainty between the values 0.3717 and 0.6101. Regarding the F1 score, this performance value is with 95% certainty between the values 0.1463 and 0.3354. Since both the interval ranges are rather large, again this indicates the performance of the model is too dependent on the training and test set. This makes the model performance unreliable.

Accuracy = [0.3717, 0.6101] F1 score = [0.1463, 0.3354]

7.3 Artificial neural network

The final selected features for the artificial neural network algorithm are not based on a specific neural network feature selection algorithm. The selection is essentially based on the most important features resulting from the feature selection algorithms based on a random forest, explained in Section 5.3.7. These are consistent with the classification approach of human operators, since many of the most important features contain the historic behaviour of the vessel. This historic behaviour is also the focus of a human operator during classifying. Another aspect that is considered during selecting the features is to exclude highly correlated features. In concrete terms, this means simply one speed, acceleration, heading and angular speed related feature will be included. Moreover, either a heading or course over ground related feature may be included, since their meanings are too similar. Due to the known relatively high run time of a neural network, the amount of input features is maintained low. Which is why the features indicating in what area the vessel is located, are not used as input features for the neural network. This has resulted in the following selected features.

- *draught*
- *heading average*
- *acceleration sd*
- *speed max*
- *known destination*
- *angular speed sd*

The optimal parameter values resulting from the grid search approach are shown in Table 7.3, considering both optimising the accuracy and F1 score. The corresponding cross validation performance values are 0.455 (+/-0.093) and 0.402 (+/-0.044) for the accuracy and F1 score, respectively.

TABLE 7.3: Optimal tuned parameter values for each optimised performance measure (artificial neural network)

	Accuracy	**F1 score**
hidden_layer_sizes	9	11
activation	relu	relu
alpha	1e-1	1e-5
max_iter	100	500
Cross validation performance	0.455 (+/-0.093)	0.402 (+/-0.044)

The cross validation results for each parameter set during the grid search are shown in Tables 10.13 and 10.14 in the Appendix for optimising the accuracy and F1 score. The parameter that has most significant influence on the model accuracy is the type of activation function. The rectified linear unit activation function predominantly outperforms the other two functions logistic and tanh. The accuracy concerning this activation function also seems to be more robust, since the maximum difference of the mean test score is 3.1% while these are equal to 6.9% and 7.8% for the logistic and hyperbolic tangent, respectively. The biggest difference between the average model accuracies for this exact implementation is 8.1%. No firm conclusion can be made regarding the parameter value alpha. Analysing the mean accuracy with

varying alpha values and all other values fixed shows an alternation between which alpha value is optimal. The number of neurons in the hidden layer neither shows any obvious relation with the accuracy performance. For example, increasing the number of neurons is not equivalent to better performance or vice versa. Neither is there an amount of neurons that significantly outperforms the other amounts. Furthermore, the maximum number of iterations does not have a significant effect on the performance. The best performing parameter values regarding optimising the accuracy are used to train the artificial neural network and, subsequently, test the model performance on the independent test set. The corresponding performance involves a classification report and confusion matrix, which can be found in Tables 7.6 and 10.15 in the Appendix, respectively. The final accuracy and F1 score based on the test set are as follows.

Accuracy = 0.4118 F1 score = 0.1058

How accurate these two results are, is evaluated using the corresponding constructed bootstrap confidence interval. The accuracy of the final artificial neural network is with 95% certainty within the range of 0.3016 and 0.5411. The performance value F1 score, however, is significantly low and is with 95% certainty within the range 0.0904 and 0.2504. Both confidence intervals are relatively large, indicating a significant dependency of the performance on the training and test set. This means the performance is inaccurate and not considered reliable enough.

Accuracy = [0.3016, 0.5411] F1 score = [0.0904, 0.2504]

7.4 Model comparison

In this section, the accuracy results of the models obtained from each of the three machine learning algorithms will be compared. This comparison between the specific algorithm results is justified, since each model is trained on the same training set as well as tested on the same test set. Due to the biased implementation of (grid search) cross validation, those resulting accuracy values are not considered during the comparison. The classification reports, shown in Table 7.4, 7.5 and 7.6, clearly illustrate which vessel type categories occur most in the test set. The HSC is not considered, since the concerning one vessel in the total dataset ended up in the training set. The majority of vessels concerns cargo ships, followed by the categories harbour vessels, other, passenger ships, tankers and dredging vessels in descending order. The least common categories are the military and fishing vessels, with a substantial minority in pleasure crafts. The categories have similar proportions in the training set, as shown in Figure 5.23, which means relatively many categories have to be predicted based on a limited amount of data. These category sizes have impact on the performance of each category, as will be analysed in more detail later this section.

TABLE 7.4: Classification report random forest

	Precision	Recall	F1 score	Support
Cargo ship	0.61	0.63	0.62	6806
Dredging vessel	0.23	0.13	0.17	855
Fishing	0.00	0.00	0.00	301
HSC	0.00	0.00	0.00	0
Harbour vessel	0.55	0.81	0.66	1807
Military vessel	0.82	0.40	0.54	437
Other	0.37	0.10	0.16	987
Passenger ship	0.08	0.05	0.06	926
Pleasure craft	0.00	0.00	0.00	12
Tanker	0.25	0.34	0.29	886
avg / total	0.49	0.50	0.48	13017

TABLE 7.5: Classification report support vector machine

	Precision	Recall	F1 score	Support
Cargo ship	0.62	0.80	0.70	6806
Dredging vessel	0.29	0.29	0.29	855
Fishing	0.00	0.00	0.00	301
Harbour vessel	0.75	0.46	0.57	1807
Military vessel	0.49	0.69	0.57	437
Other	0.03	0.00	0.01	987
Passenger ship	0.31	0.51	0.38	926
Pleasure craft	0.00	0.00	0.00	12
Tanker	1.00	0.08	0.14	886
avg / total	0.56	0.56	0.52	13017

TABLE 7.6: Classification report artificial neural network

	Precision	Recall	F1 score	Support
Cargo ship	0.53	0.71	0.61	6806
Dredging vessel	0.00	0.00	0.00	855
Fishing	0.00	0.00	0.00	301
Harbour vessel	0.16	0.24	0.19	1807
Military vessel	0.00	0.00	0.00	437
Other	0.23	0.10	0.14	987
Passenger ship	0.00	0.00	0.00	926
Pleasure craft	0.00	0.17	0.01	12
Tanker	0.01	0.00	0.01	886
avg / total	0.32	0.41	0.36	13017

Comparing the results of the three implemented algorithms, represented in Figure 7.1, highlights that the support vector machine is the best performing model. This is supported by both the accuracy resulting from the test set, which is significantly the highest, as from the bootstrap confidence interval, which shows the highest interval range for the support vector machine. Therefore, the trained support vector machine is chosen to be the optimal model within the scope of this research. However, the support vector machine model does not significantly outperform the other two models, based on the confidence intervals. The confidence interval of the random forest is only slightly lower and the interval of the neural network overlaps with the confidence interval of the support vector machine. The spread in all accuracy confidence intervals is around 0.23, which is a relatively wide range. Since the absolute ranges are similar, the performance of the three models is considered as equally reliable. The wide range indicates that the model performance can drastically differ if another combination of training and test set is used, which implies overfitting. Overfitting occurs because the model is fitted too accurately to the considered training set and, consequently, is not able to generalise well when testing on an independent test set. Due to the significantly small training set, the model can only learn patterns existing in the training set and, therefore, lacks other existing patterns that do exist in the real world. In other words, there is insufficient ground truth. Hence, optimising a particular training set can fail to fit a similar performance on a different dataset. This is supported by the recall values of the trained support vector machine, shown in Table 7.5 in the Appendix. The recall represents the percentage of correctly predicted instances for a certain vessel type relatively to the total number of instances of this vessel type. Both least-occurring categories, pleasure craft and fishing vessel, have a recall equal to 0, while in general the more frequently occurring categories have a higher recall value. However, this relation is not proportional, which might imply that some categories require less data to find the existing patterns. A similar conclusion can be drawn for the trained random forest and the trained neural network.

Most details about each model performance can be obtained from the confusion matrix. All three models have certain performance aspects in common, while more specific parts of the performance differ. First, the similar aspects will be evaluated based on the confusion matrices shown in Tables 10.10, 10.12 and 10.15 in the Appendix. All models have difficulties predicting the less occurring category fishing. This

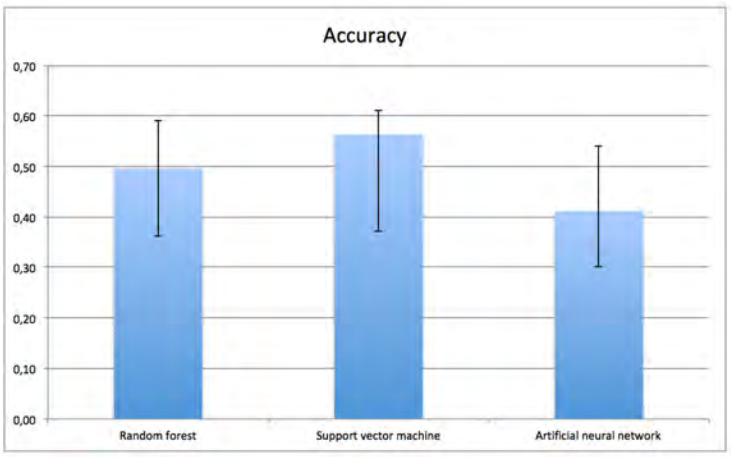

FIGURE 7.1: Performance comparison with each final model

category is never predicted due to the lack in amount of data. For both the random forest and support vector machine the category pleasure craft is never predicted as well. Considering the neural network, this occurs for the categories military vessels and passenger ships, besides the fishing vessels. This is explained by the fact that the categories with least amount of unique vessels are HSC, pleasure craft, fishing, passenger ship and military vessel, starting with the smallest category as is shown in Figure 5.23. Moreover, a remarkable fact is that for all models the category cargo is predicted the most. Since this is the most occurring vessel type, this should be true. However, it appears that cargo is predicted for every category when the actual category is not cargo. Such a tendency, to predict the most occurring category more frequently than necessary, is often a result from an unbalanced dataset. A similar conclusion, yet to a smaller extent, can be drawn for the category harbour vessel. This category is often predicted while the actual category differs. Finally, even though the category tanker has relatively much data, this category is poorly predicted. For each of the three models the majority of the tankers is predicted as cargo, which is a reason to believe that tankers and cargo are not distinctive enough. Based on domain knowledge this was a reasonable expectation, since both the speed behaviour and the size of tankers and cargo ships are similar. A possible solution would be to combine the two categories into one category during further research. Overall the confusion matrices indicate that the determined vessel type categories are not distinctive enough for this particular dataset. This seems especially the result of the unbalanced dataset in combination with the significant deficiency in data of the least occurring categories.

Regarding the three confusion matrices individually, more detailed conclusions can be drawn. These will be clarified in the next three paragraphs. The discriminating capability of the trained support vector machine is the highest. As the random forest model and the neural network model both have many miss-classifications for a significant number of categories, the support vector machine has considerably less miss-classifications. More specifically, most of the classifications of the support vector machine model are either on the diagonal of the confusion matrix, indicating TPs, or are predicted as cargo, which is shown in the corresponding confusion matrix in Figure 10.12. Both the random forest model and neural network model show more variation in this regard. The categories with a majority of instances rightly predicted are cargo ship, harbour vessel, military vessel and passenger ship. The majority of instances of the remaining categories are predicted as cargo. This means identifying the patterns for the categories dredging, fishing, other, pleasure craft and tanker is a difficult task for the support vector machine, which makes it generally predict the most occurring category instead. For this reason, the least common categories pleasure craft and fishing are always predicted as a cargo ship.

The trained random forest model correctly classifies the majority of the instances of the categories cargo ship, harbour vessel and military vessel, which is shown in the corresponding confusion matrix in Figure 10.10. Compared to the support vector machine, it does not include the category passenger ship. This category is mostly predicted as a harbour vessel. For all remaining categories, again, is the majority of the instances miss-classified as a cargo ship. Another difference compared to the performance of the trained support vector machine is the larger variation in miss-classified categories. For example, more instances are predicted as a harbour vessel, when it is actually one of the other categories. Another example involves that the military vessels are only predicted as military or cargo based on the support vector

machine, but based on the random forest model the military vessels are predicted as military, cargo, harbour, other, passenger and tanker. This indicates that the obtained random forest is less able to strictly separate these vessel types compared to the obtained support vector machine.

The trained neural network model performs significantly the worst. The single one category that correctly predicts the majority of instances is the category cargo ship, which is shown in the corresponding confusion matrix in Figure 10.15. The majority of passenger ship instances is predicted as harbour vessel, the majority of military vessel instances is predicted as the category other and the majority of instances of the remaining categories is predicted as cargo ship. Besides the categories fishing, military and passenger, which the model did not predict at all, the category dredging vessel is hardly predicted. The poor performance may be a result of the predetermined number of hidden layers, which is set to one. Therefore, it is advised to experience with the number of hidden layers during further research, since more hidden layers could cause significant better performance values. Moreover, the final selected input features do not contain any information about the location of the vessel, which is considered as essential indicator for the human operator to make the classification. However, the area type related features are not highly ranked in the implemented feature selection algorithms, since the valuable information is covered enough in other features like the sailing speed and course. Therefore, it is debatable whether this causes the lower performance compared to the random forest and support vector machine.

The parameter robustness analysis requires a more detailed clarification. The maximum differences in cross validation accuracy values of 5.2%, 13.5% and 8.1%, mentioned respectively in Sections 7.1, 7.2 and 7.3, do not consider a comparison that is entirely fair. First of all, the neural network model had more parameters to be tuned. Secondly, certain parameters are implemented by varying more values than other parameters. In concrete terms, this means the parameter hidden_layer_sizes is varied trying the seven values 4, 6, 7, 8, 9, 10 and 11, while the parameter alpha is varied trying only the three values 1e-5, 1e-3 and 1e-1. Moreover, the parameters of the different models are hard to compare. Implementing a random forest with varying number of trees between 50 and 100 is completely different than implementing a neural network with varying number of neurons between 50 and 100. A conclusion that can be drawn, considering the predetermined range of varying parameter values, is that the random forest model is most robust. However, the effect of varying parameter values on the performance is significant, which means a change in parameter value is unwanted.

Finally, all three algorithms are easy to use with the implementation of Scikit-learn, once the appropriate knowledge about the algorithms and the online documentation about the implementation of the algorithms are used. Intuitively the neural network algorithm is the hardest to interpret. The exact consequence of tuning the parameter values is often perceived as a black box, partly due to the chosen activation function. Similarly, the interpretation of the support vector machine is rather complex due to the possible high dimensional feature space and due to implicitly transforming the features into an even higher dimensional space. The random forest algorithm is intuitively the easiest algorithm to understand, since it is a rule-based algorithm. Considering the ease of use and the business context of this problem at Thales, all three algorithms are suitable for use during further research on classifying vessel types.

The trained support vector machine is chosen as optimal model based on the previous results and evaluation. The performance of the support vector machine, given the limited amount of data, is hopeful towards acquiring better performance with a relatively much larger dataset. The potential to develop a classification model for the Navies is reflected in the fact that the support vector machine model is able to make the least amount of miss-classifications along with the least amount of variation in miss-classifications.

8 Discussion

The aim of this master thesis was to investigate whether it is possible to develop a sufficiently reliable product to classify vessel types using machine learning. This research has experienced challenges and several limitations towards achieving better results.

One challenge of this research is data availability. Data is sparse within the naval field, which means it requires cooperation with trusted stakeholders of the organisation or a large amount of money in order to obtain AIS data. There are no open source databases containing AIS data. Due to Thaless collaborations with outside parties, the used dataset was acquired for research purposes. Another challenge regarding the classification of vessel types is the unbalanced worldwide vessel composition. Since several vessel types around the world dominate the majority of the vessels, these are over-represented within each dataset, which makes it harder to correctly predict the less common vessel types.

Furthermore, this research has come across three fundamental limitations, which may have distorted or affected the results. First of all, the acquired dataset contains a limited amount of data. This is caused by the length of the time interval in which the AIS messages are collected. Acquiring 40 minutes of AIS messages during the day can only cover a certain, relatively small, amount of vessels. This leads to a limited number of observations for certain vessel types, which means the ground truth of these types is scarce. It does not only cause difficulties finding the existing patterns between the variables and these vessel types, it also has a clear impact on the macro-average F1 score. If there is only a limited amount of data of a certain vessel type it is highly likely this specific vessel type will not be included in the test set. Consequently, during the test phase, this vessel type may cause the corresponding precision or recall and, therefore, the F1 score to be equal to 0. This specific F1 score of 0 has a significant impact on the macro-average F1 score, causing it to decrease. The limited amount of data also ensures the high dependency of the model performance on the specific training and test set.

Another limitation is the small and the specific geographical area where the AIS messages originated from. Most of the vessels in the dataset are located in the harbour of Rotterdam, which has two undesirable consequences. The first consequence is that a large amount of vessels are not sailing or are sailing with the same maximum speed, which means the data does not contain the vessels behaviour or its behaviour is not distinguishable from the other vessels. However, this data is deliberately kept in the dataset, since the size of the dataset is already far too small. Consequently, it does sabotage the performance of the final model. The second consequence involves the poor representation of the vessels that are generally sailing on the North Sea. A harbour attracts certain vessel types, which may not even appear on the North Sea. This restricted area predominantly containing vessels in the harbour affects the results negatively, since the vessels located close to a harbour are not relevant to the practical use of the desired product.

The previous two limitations together cause a certain wrong representation of the vessel composition on the North Sea. Since the North Sea is the focus of the desired product, these limitations make the second mentioned challenge even more challenging. The wrong representation of the vessel composition will influence the unbalanced worldwide vessel composition and skew the dataset even more.

The final limitation involves the large amount of missing values for certain features in the dataset. Since the implemented algorithms cannot deal with missing values, these need to be imputed. Imputing missing values is not desirable, since it concerns a probability that the imputed value is not equivalent to the true value. This has been especially a high-risk concerning the variable draught. It is considered as the most important variable for the random forest algorithm. Imputing the many missing values will shape a dataset that is not similar to the actual reality and can, as a result, develop wrong patterns between the variables during the training phase.

In general, the inadequate results could be due to three different factors, which are the data, the implemented techniques or generally machine learning as an approach. The above-mentioned limitations

indicate that the specific dataset is a good possibility to be the limiting factor for developing a high performing classification model. Which would imply that AIS data in general is appropriate. However, it may also be the specific implemented machine learning techniques impeding the required level of performance. This research did only implement three algorithms, while the best-suited algorithm for this specific problem may be a different algorithm. The final factor involves the unsuitability of machine learning as approach. However, this can only be established if all possible classification algorithms are implemented using a high quality and, therefore, representative dataset. Implementing this during further research should conclude whether machine learning as approach is appropriate or not for this problem.

Finally, this section contains a remark to keep in mind during further research. This research computes the accuracy based on the percentage of correctly predicted AIS message updates. However, for the human operator on board of a naval vessel, it is far more interesting to know the percentage of correctly predicted unique vessels. This is a rather difficult issue, since a part of a specific vessel its instances may be predicted correctly, while the remaining instances are not. This issue should be considered during further research at Thales, in close collaboration with human operators in order to contribute to the best support for the human operator.

9 Conclusions

This section answers the research question and sub questions presented in the research description in Section 3.2. Each sub question is answered individually, having a contribution to answering the main question of this research.

Is the quality of the data sufficient?
The quality of the data is evaluated based on two approaches, a priori and a posteriori. The a priori data quality is analysed before using the data for training the models. During the data analysis in Section 5.2 several aspects of the data are observed that may be challenging. The three considered aspects are the small time frame, the specific geographical location and the amount of missing values in certain variables. The first two aspects, time and geographical location, are the cause of limited amount of data and the exact composition of the vessel types. The specific and small time frame only covers a snapshot and, therefore, a limited amount of vessels during the day. The specific geographical location predominantly provides a selection of vessel types close to the harbour of Rotterdam, which indicates many vessels without any sailing speed. Moreover, this involves a different composition of vessel types compared to non-harbour areas. Consequently, these two aspects cause a misleading picture of the vessel composition during Navy missions on the North Sea. The third aspect involves relatively high amount of missing values in certain variables. Especially the variable draught is essential for predicting the vessel type and contains many missing values, which influences the model performance. At the same time, the dataset contains similar data that the operator uses to manually classify the vessel types, which indicates potential towards the variables predictive power.

The a posteriori data quality is analysed based on the model performance values. The results in Section 7 confirm the suspicion of the challenges, since it shows a significantly low predictive power of the models towards predicting the vessel type. The highest accuracy, originating from the support vector machine, is equal to 0.5636. Especially the uneven distribution of the categories combined with the limited amount of data seems to be a major issue. This makes identifying patterns between variables hard, which results in not being able to predict the least occurring categories and, therefore, many miss-classifications. Moreover, the determined vessel type categories are not distinctive enough, which follows from the confusion matrices. The small training dataset causes overfitting, which is confirmed by the bootstrap confidence intervals. Hence, all these issues together indicate that the data quality of this particular dataset is not considered sufficient. However, AIS in general does have potential to achieve high classification results, since the similar research of Ljunggren (2017) [19] does achieve an accuracy value of about 0.92 for the ensemble method applied to dataset B.

What validation criteria apply best for this specific problem?
Due to the very nature of the dataset, two validation criteria are considered during the research of this classification problem. Since the dataset is unbalanced, the macro-average F1 score is a wel suited validation measure. It gives equal importance to each category of the unevenly distributed target variable. The worldwide vessel composition contains a majority of cargo vessels. However, the less common vessel types are at least as important, meaning such a desired classification model should be able to properly manage this aspect. Especially the less occurring vessel types are often a target during a mission of the marine, like detecting drug smugglers or illegal fishing. Therefore, particularly these vessel types need greater importance to be accurately classified. The macro-average F1 score satisfies this requirement, which is why the macro-average F1 score fits the classification problem within the naval context best.

However, the main focus during this research was optimising the accuracy, because the performance of the model using this dataset is in general low. In order to make this classification problem suitable for developing a product within the organisation, first the overall performance needs to be increased. Such an overall performance evaluation measure is the accuracy, since it measures how good the predictions are on average. Once the accuracy is confirmed high enough, the macro-average F1 score should be the primary evaluation measure. A sufficient accuracy is equivalent to an accuracy of at least 0.9, since this is achieved in the literature [19] and indicates a good overall performance. Moreover, the results show that, when improving the F1 score, it is more likely the accuracy has a reasonable performance level as well than the other way around. In other words, if the accuracy is optimised, the particular focus can be

directed towards simply correctly predicting the most occurring vessel type. This could cause a relatively high accuracy while the F1 score is low. For this particular problem it seems that the accuracy remains at the same level when the macro-average F1 score is optimised. Therefore, switching from optimising the accuracy to optimising the F1 score once the accuracy is high enough is reasonable, because it will not result in a significant decrease in overall performance and it will focus more on the less common categories. Note that the absence of a significant decrease in accuracy, when optimising the F1 score, is not necessarily true for each classification problem.

Given the available data, which machine learning techniques apply best to classify vessels?
Out of the three implemented algorithms, the support vector machine has the best classification performance, considering this data and the implemented parameter values. Section 7.4 shows the model reaches an accuracy of 0.5636. The performance of the artificial neural network is significantly lower with an accuracy of 0.4118 and the random forest its accuracy is equal to 0.4960. Even though the trained support vector machine performs the best, it is not considered adequate towards a sufficiently reliable product due to the lacking predictive power originating from the properties of the dataset. Besides the rather low performance, the bootstrap confidence interval of [0.3717, 0.6101] indicates an insufficient level of the reliability of the model performance. The wide 95% confidence range signifies that the model performance is strongly dependent on the combination of chosen training and test set, which means the performance robustness does not coincide with the appropriate and desired level. However, this research shows that the support vector machine has the highest performance values, which means that model is best applied to classify the vessels using the given dataset.

How robust are the performance values of the models obtained by the machine learning techniques, with regard to the parameter values and the geographical dependence?
On average the results of the grid search in Section 7 seem to provide robust mean performance values. The random forest performance values with varying sets of parameter values is considered most robust, since the maximum mean accuracy difference is equal to 5.2%. Concerning the mean accuracy values of the final support vector machine and artificial neural network, these maximum differences are equal to 13.5% and 8.1%, respectively. Although the percentages are relatively low on a scale of 1 to 100, these represent the mean difference in performance values. Therefore, the exact performance of one model tested on a certain test set may differ significantly from another model tested on this set. Changing a parameter value within the considered parameter value sets does not provide a significantly worse performing model, but the performance cannot be considered as completely robust. Clearly these results depend on the predetermined parameter value sets. Meaning, different parameter values outside the considered range of values, could result in a significant decrease of the performance. In reality, it is preferred to implement such a model specifically designed and optimised for a certain customer.

Due to time limitation of the research, the robustness of the performance values towards the geographical dependence has not been tested. However, it is recommended to test the plausible dependency, since this is essential for determining whether future customers should arrange the provision of quality data themselves or whether the use of similar data is sufficient.

What skills and knowledge should be present in the organisation to develop the desired product?
In order to develop the desired product, the employees responsible for implementing the product should be familiar with the whole process this thesis has guided. More concrete, the general machine learning approach should be known to all, with more specialised knowledge of each aspect divided between qualified employees. Such aspects are domain knowledge, data analysis, data pre-processing, mathematical knowledge behind the machine learning algorithms, applied programming skills and the TACTICOS environment that should incorporate the desired product. Moreover, it is important to have employees supervise the whole process, who are able to know the consequences of changes in, for example, the data or parameter values. Especially theoretical knowledge about machine learning is essential to support and confirm decisions made during the development of such models. The domain knowledge is essential in sense of how to tackle the discovered machine learning challenges in this particular context. As all aspects are available in the organisation, one better represented than the other, the biggest challenge is to combine all relevant theoretical knowledge and practical knowledge in order to being able to develop the desired product.

Based on these sub questions, the main research question will be answered in the highlighted box below.

Is it possible to develop a sufficiently reliable product that classifies vessels using machine learning?

Considering all previous sub questions, the answer to this research question is twofold. The results of the implementation during this research show that the accuracy performance of the best-optimised model has the significantly low value of 0.5636 and is, therefore, not of a sufficient level towards the desired product. More specifically, the final model would have been considered sufficiently reliable if the accuracy was around 0.9 or higher. Moreover, it also depends on the macro-average F1 score and how accurately the essential less common categories are predicted. This aspect is even worse, since the less common categories fishing and pleasure craft are not predicted at all. This indicates a low discriminating capability between the vessel type categories as determined in Section 5.3.6. The final model is not only considered insufficient in terms of the test accuracy, but also in relation to the lack of robustness, which the model requires in order to develop a sufficiently reliable product. In other words, considering the same model structure, but trained and tested on a different part of the dataset, does provide significantly different performing models. The model shows a big dependency on this specific dataset, which is at least due to the cause of far too little amount of data. Implementing the final support vector machine model as support for the operator would only distract him from the classification, since it frequently suggests wrong and, therefore, unusable predictions.

Despite the specific implementation of the classification problem, this research still shows potential towards the possibility to develop the desired product. This is supported by internal experiments of a Thales innovation team. The experiments make use of data over a larger time span of approximately 18 days, which produces significantly better results. Therefore, the desired product may be possible with more and better represented training data. The answer of the research question concerning this research is negative. However, implementing the crucial recommendations in future research, mentioned in Section 10, will have to establish whether machine learning is suitable to develop a sufficiently reliable product to classify vessels.

10 Recommendations

Improvements can be made for a number of aspects in this research. Since the aim of this research is to investigate whether there is potential in developing a classification product for the Navies, the focus of the recommendations is towards the required steps to achieve this level. First, this section discusses crucial issues that ensure most potential to reach the desired classification model. Subsequently, less essential, more detailed issues will be recommended.

10.1 Main recommendations

As discussed in Section 8 Discussion, the inadequate results can be caused by the specific dataset, the implemented techniques or generally machine learning as an approach. Since examining the latter would mean considering every possible model, the most essential recommendations are focused on the first two factors.

Most crucially, significant performance improvement steps can be achieved in terms of the properties of the dataset. The key recommendation is to use a dataset that is representative for the real life situations that will occur when using the model in the required area. More specifically, for this particular problem it concerns specific focus for the following aspects of the dataset.

Above all, the dataset should contain incoming messages over a range of multiple days. Because, consequently, enough data for each vessel type will be available to discover patterns and to exclude dependency in the data that would occur due to patterns of a specific day or time. This results in a better understanding of the general daily vessel composition in that area. The implemented dataset covers only a snapshot of the daily vessels due to the small time period and, therefore, an undersized amount of updates of certain target categories, which is not enough for a high performance.

Secondly, the dataset should contain incoming messages that cover the geographical location more broadly, if a reduced geographical dependence of the model is desired. The training dataset should represent the entire relevant area for implementing the model during missions. Since the focus during missions of the Dutch Navy is on the entire North Sea, this should be represented in the dataset to train the model accurately. A major disadvantage of the implemented dataset is the fact it mainly covers a harbour area instead of the entire North Sea. First of all, this is only a limited part of the geographical location the dataset should contain. But, more importantly, vessels in a harbour area are less relevant during missions. Moreover, the majority of the vessels located in a harbour are anchored or sailing with the same speed, which means no distinctive patterns can be determined based on the two most characteristic features vessel speed and location. These vessels do not contribute to the discriminating capability and, consequently, are responsible for a decrease in data quality. Therefore, it is recommended not to include vessels located in a harbour in the training dataset if the remaining size of the dataset is sufficient.

The previous two aspects indirectly should result in a dataset that is representative for the composition of vessel types in the preferred area. Apart from ensuring a complete overview of the vessels in that area, it ensures the less occurring vessel types have enough ground truth and, therefore, to be sufficiently represented. This is due to the increasing amount of data by considering message update of several days instead of only 40 minutes. Consequently, a higher discriminating capability of the vessel type categories must be achieved.

Moreover, an essential aspect of the dataset is how the many subtypes of the vessels are combined. As mentioned in Section 5.3.6, vessel types should be combined based on similarities in the characteristics (variables) in order to create a model with high discriminating capability. Apart from this, the implemented composition of these sub types must be relevant for the implementation of the model in practice. Varying the composition of the categories of the target variable during experiments has shown the significance on the model performance, which can also be concluded to some extent from the research of Ljunggren (2017) [19]. The optimised models of Ljunggren show that considering dataset B, containing only the four biggest vessel type categories, does achieve a significantly better performance compared

to dataset A, containing eleven categories. The accuracy resulting from dataset B is equal to 0.92, when dataset A only achieves an accuracy of 0.75. Consequently, it can be concluded that less vessel type categories increases the predictive power, if the biggest categories are considered. However, it is not allowed to remove certain vessel types from the dataset to increase the discriminating capability without further justification. The model should be able to deal with all kinds of incoming message updates, which means certain vessel types cannot be neglected. It is recommended to do more research into achieving higher discriminating capability based on varying the vessel type composition.

Implementing a bigger sized dataset increases the probability to find patterns between the variables and the target variable, which will likely result in a higher predictive power. The implementation of a bigger sized dataset still requires experiments to find the most optimal and appropriate data pre-processing steps and model optimisations steps. Most important is to efficiently combine the domain knowledge with the available skills to being able to develop a product that provides useful support for the operators. Useful is to be defined by the exact mission of the marine vessel including a required certainty of the model predictions. A concrete example of combining domain knowledge with the available skills is to engineer features based on domain knowledge, which supports the model to make a distinction between the vessel types.

Apart from using a dataset that is representative for the required implementation, other machine learning techniques may also succeed in achieving a better model performance. This cannot be concluded from this research, yet experiments with other machine learning techniques should determine which algorithm suits this classification problem best. Therefore, a second important recommendation for further research is to experiment with other machine learning techniques, such as gradient boosting, logistic regression, naive Bayes and k-nearest neighbour, using enough ground truth data. Using the same dataset, the results of the three techniques random forest, support vector machine and neural network are appropriate for a proper comparison.

10.2 Additional recommendations

During this research various issues are observed that require more attention. Due to the time limitations and determined scope of the research these have not been implemented, but instead are included in this section. The following recommendations for future research cover the implementation of experiments relevant to create a complete view of useful methods. The experiments can either clarify which implementations can be excluded or which implementations actually improve the model performance and could increase the potential to develop a product.

The first recommendation involves experiments to test the geographical dependence of the obtained model, which is described in Section 6.4.4 but was not implemented due to the restricted time. These experiments should determine the effect of a different geographical location towards the model performance. The result is essential to determine whether the navies should deliver data to Thales from the specific location or whether a comparable training dataset can be used. Moreover, it indicates whether the same model can be used or whether different models or composition of models should be used.

Another recommendation, which will contribute to increasing reliability of the final product, is the implementation in the TACTICOS system of a filter system prior to using the obtained model. This should filter out all vessel updates that show irregular behaviour. Not considering these vessels in the classification model will increase the focus towards the normal behaving vessels, which predictions should support the human operator. The abnormal behaving vessels need special attention of the human operator regardless and can, therefore, be filtered out. The exact determination of an irregular behaving vessel is a rather complex task, which will not be discussed here and is open for discussion at Thales.

The results of this research are restricted due to the selection of implemented parameter values. Therefore, it is recommended to expand the iterative process with more experiments, including more variation in parameter values and implementing different selections of features. More specifically, especially the artificial neural network requires experiments that should test whether a more complex model results in better performance. The number of hidden layers and input features were restricted to a low number due to the expected long running times. However, it is recommended to implement multiple hidden layers, which should determine whether applying deep learning methods increase the model performance. Moreover, the neural network should be implemented with more selected input features. Especially the

effect of area type related features need attention, since this is not accurately examined during this research.

Since some features have a significant amount of missing values, the imputation method may have significant influence on the model performance. Therefore, it is recommended to consider using external databases to impute the feature value, since these values have a real chance to be equal to the true value. Clearly, this imputation method is restricted to features that are known in external databases like marinetraffic.com and digital-seas.com.

Finally, it is recommended to continue the feature engineering process consistent with domain knowledge of the human operators. Especially the characteristics of the vessels, like the size of the vessel, may be better represented in the dataset. External databases like marinetraffic.com and digital-seas.com may contain such relevant additional information. Moreover, other features representing the behaviour of the vessel may contain significant predictive power. A list of potential features to be engineered, derived from literature, is added to the Appendix in Table 10.16.

Appendix

FIGURE 10.1: TACTICOS console

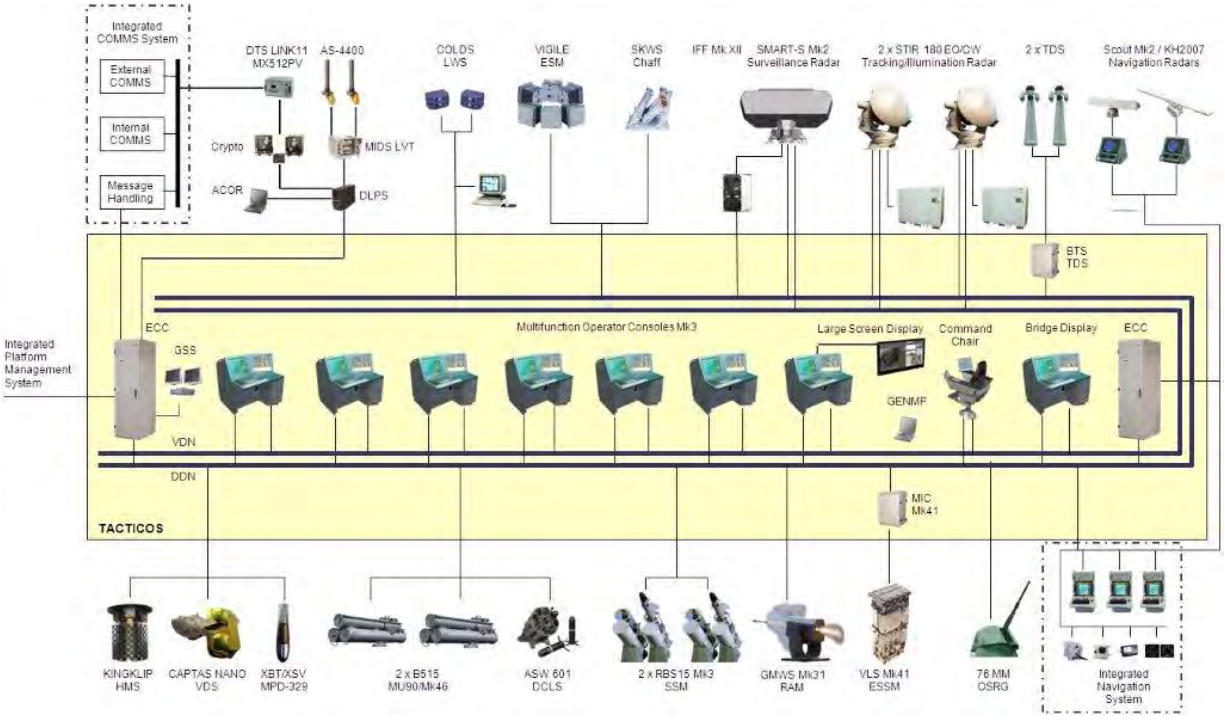

FIGURE 10.2: TACTICOS specifications

TABLE 10.1: Dynamic data AIS message

CLASS A AIS POSITION REPORT (MESSAGES 1, 2, AND 3)

A Class A AIS unit broadcasts the following information every 2 to 10 seconds while underway, and every 3 minutes while at anchor at a power level of 12.5 watts.

Parameter	Bits	Description
Message ID	6	Identifier for this message 1, 2 or 3
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated. See Section 4.6.1, Annex 2; 0-3; 0 = default; 3 = do not repeat any more
User ID	30	MMSI number
Navigational status	4	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 ships 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 ships 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)
Rate of turn ROT_{AIS}	8	0 to +126 = turning right at up to 708 deg per min or higher 0 to -126 = turning left at up to 708 deg per min or higher Values between 0 and 708 deg per min coded by $ROT_{AIS} = 4.733\ SQRT(ROT_{sensor})$ degrees per min where ROT_{sensor} is the Rate of Turn as input by an external Rate of Turn Indicator (TI). ROT_{AIS} is rounded to the nearest integer value. +127 = turning right at more than 5 deg per 30 s (No TI available) -127 = turning left at more than 5 deg per 30 s (No TI available) -128 (80 hex) indicates no turn information available (default). ROT data should not be derived from COG information.
SOG	10	Speed over ground in 1/10 knot steps (0-102.2 knots) 1 023 = not available, 1 022 = 102.2 knots or higher
Position accuracy	1	The position accuracy (PA) flag should be determined in accordance with the table below: 1 = high (<= 10 m) 0 = low (> 10 m) 0 = default
Longitude	28	Longitude in 1/10 000 min (+/-180 deg, East = positive (as per 2's complement), West = negative (as per 2's complement). 181= (6791AC0h) = not available = default)
Latitude	27	Latitude in 1/10 000 min (+/-90 deg, North = positive (as per 2's complement), South = negative (as per 2's complement). 91deg (3412140h) = not available = default)
COG	12	Course over ground in 1/10 = (0-3599). 3600 (E10h) = not available = default. 3 601-4 095 should not be used
True heading	9	Degrees (0-359) (511 indicates not available = default)
Time stamp	6	UTC second when the report was generated by the electronic position system (EPFS) (0-59, or 60 if time stamp is not available, which should also be the default value, or 61 if positioning system is in manual input mode, or 62 if electronic position fixing system operates in estimated (dead reckoning) mode, or 63 if the positioning system is inoperative)
special maneuvre indicator	2	0 = not available = default 1 = not engaged in special maneuver 2 = engaged in special maneuver (i.e.: regional passing arrangement on Inland Waterway)
Spare	3	Not used. Should be set to zero. Reserved for future use.
RAIM-flag	1	Receiver autonomous integrity monitoring (RAIM) flag of electronic position fixing device; 0 = RAIM not in use = default; 1 = RAIM in use. See Table
Communication state (see below)	19	See Rec. ITU-R M.1371-5 Table 49
Number of bits	168	

TABLE 10.2: Static data AIS message

AIS CLASS A SHIP STATIC AND VOYAGE RELATED DATA (MESSAGE 5)

In addition, the Class A AIS unit broadcasts the following information every 6 minutes. Should only be used by Class A shipborne and SAR aircraft AIS stations when reporting static or voyage related data:

Parameter	Bits	Description
Message ID	6	Identifier for this Message
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated. Refer to §?4.6.1, Annex 2; 0-3; 0 = default; 3 = do not repeat any more
User ID	30	MMSI number
AIS version indicator	2	0 = station compliant with Recommendation ITU-R M.1371-1 1 = station compliant with Recommendation ITU-R M.1371-3 (or later) 2 = station compliant with Recommendation ITU-R M.1371-5 (or later) 3 = station compliant with future editions
IMO number	30	0 = not available = default – Not applicable to SAR aircraft 0000000001-0000999999 not used 0001000000-0009999999 = valid IMO number; 0010000000-1073741823 = official flag state number.
Call sign	42	7?=?6 bit ASCII characters, @@@@@@@ = not available = default Craft associated with a parent vessel, should use "A" followed by the last 6 digits of the MMSI of the parent vessel. Examples of these craft include towed vessels, rescue boats, tenders, lifeboats and liferafts.
Name	120	Maximum 20 characters 6 bit ASCII "@@@@@@@@@@@@@@@@@@@@" = not available = default The Name should be as shown on the station radio license. For SAR aircraft, it should be set to "SAR AIRCRAFT NNNNNNN" where NNNNNNN equals the aircraft registration number.
Type of ship and cargo type	8	0 = not available or no ship = default 1-99 = as defined below 100-199 = reserved, for regional use 200-255 = reserved, for future use Not applicable to SAR aircraft
Overall dimension/ reference for position	30	Reference point for reported position. Also indicates the dimension of ship (m) (see below) For SAR aircraft, the use of this field may be decided by the responsible administration. If used it should indicate the maximum dimensions of the craft. As default should A = B = C = D be set to "0"
Type of electronic position fixing device	4	0 = undefined (default) 1 = GPS 2 = GLONASS 3 = combined GPS/GLONASS 4 = Loran-C 5 = Chayka 6 = integrated navigation system 7 = surveyed 8 = Galileo, 9-14 = not used 15 = internal GNSS
ETA	20	Estimated time of arrival; MMDDHHMM UTC Bits 19-16: month; 1-12; 0 = not available = default Bits 15-11: day; 1-31; 0 = not available = default Bits 10-6: hour; 0-23; 24 = not available = default Bits 5-0: minute; 0-59; 60 = not available = default For SAR aircraft, the use of this field may be decided by the responsible administration
Maximum present static draught	8	In 1/10 m, 255 = draught 25.5 m or greater, 0 = not available = default; in accordance with IMO Resolution A.851 Not applicable to SAR aircraft, should be set to 0
Destination	120	Maximum 20 characters using 6-bit ASCII; @@@@@@@@@@@@@@@@@@@@ = not available For SAR aircraft, the use of this field may be decided by the responsible administration
DTE	1	Data terminal equipment (DTE) ready (0 = available, 1 = not available = default)
Spare	1	Spare. Not used. Should be set to zero. Reserved for future use.
Number of bits	424	Occupies 2 slots

TABLE 10.3: Vessel sub types

Type of ship

***** (Vessels operating in U.S. waters should encode their ship type as denoted in the USCG AIS Encoding Guide) *****

Identifiers To Be Used By Ships To Report Their Type	
Identifier No.	**Special craft**
50	Pilot vessel
51	Search and rescue vessels
52	Tugs
53	Port tenders
54	Vessels with anti-pollution facilities or equipment
55	Law enforcement vessels
56	Spare - for assignments to local vessels
57	Spare - for assignments to local vessels
58	Medical transports (as defined in the 1949 Geneva Conventions and Additional Protocols)
59	Ships and aircraft of States not parties to an armed conflict

Identifiers to be used by ships to report their type			
Other ships			
First digit(1)	**Second digit(1)**	**First digit(1)**	**Second digit(1)**
1 - Reserved for future use	0 - All ships of this type	-	0 - Fishing
2 - WIG	1 - Carrying DG, HS, or MP, IMO hazard or pollutant category X	-	1 - Towing
3 - See right column	2 - Carrying DG, HS, or MP, IMO hazard or pollutant category Y	3 - Vessel	2 - Towing and length of the tow exceeds 200 m or breadth exceeds 25 m
4 - HSC	3 - Carrying DG, HS, or MP, IMO hazard or pollutant category Z	-	3 - Engaged in dredging or underwater operations
5 - See above	4 - Carrying DG, HS, or MP, IMO hazard or pollutant category OS	-	4 - Engaged in diving operations
	5 - Reserved for future use	-	5 - Engaged in military operations
6 - Passenger ships	6 - Reserved for future use	-	6 - Sailing
7 - Cargo ships	7 - Reserved for future use	-	7 - Pleasure craft
8 - Tanker(s)	8 - Reserved for future use	-	8 - Reserved for future use
9 - Other types of ship	9 - No additional information	-	9 - Reserved for future use

(1) The identifier should be constructed by selecting the appropriate first and second digits.
The second digits 1, 2, 3 and 4 reflecting categories X, Y, Z and OS formerly were categories A, B, C and D.

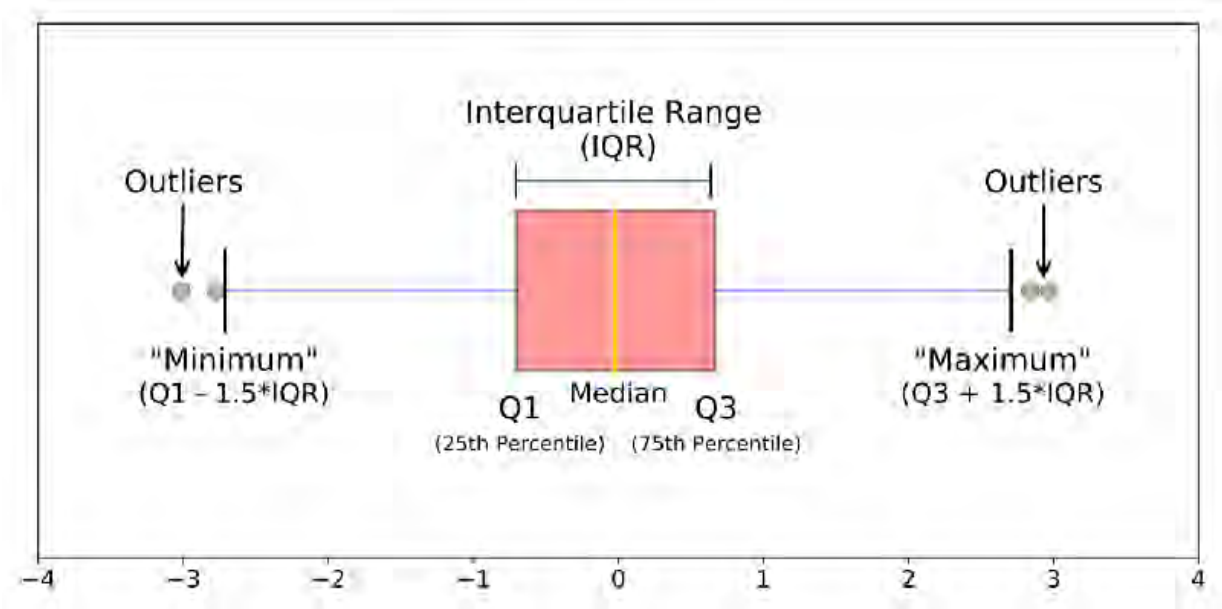

FIGURE 10.3: Interpretation of boxplot

Median = 50^{th} percentile
First quantile (Q1) = 25^{th} percentile
Third quantile (Q3) = 75^{th} percentile
Interquartile range (IQR) = 25^{th} to 75^{th} percentile
Minimum = Q1 1.5*IQR
Maximum = Q3 + 1.5*IQR
Outliers = outside the minimum-maximum range

TABLE 10.4: Original vessel types conversion

Original vessel type	Combination 1	Final combination
[1] 'Cargo ship, Category OS'	[1] 'Cargo ship'	'Cargo ship'
[2] 'Cargo ship, Category X'	[1] 'Cargo ship'	'Cargo ship'
[3] 'Cargo ship, Category Y'	[1] 'Cargo ship'	'Cargo ship'
[4] 'Cargo ship, Category Z'	[1] 'Cargo ship'	'Cargo ship'
[5] 'Cargo ship, General'	[1] 'Cargo ship'	'Cargo ship'
[6] 'Cargo ship, No additional information'	[1] 'Cargo ship'	'Cargo ship'
[7] 'Cargo ship, Reserved for future use'	[1] 'Cargo ship'	'Cargo ship'
[8] 'HSC, No additional information'	[6] 'HSC'	'HSC'
[9] 'Law enforcement vessel'	[7] 'Law enforcement vessel'	'Military vessel'
[10] 'Not available / no ship'	[8] 'Other'	'Other'
[11] 'Not specified 1'	[8] 'Other'	'Other'
[12] 'Not specified 2'	[8] 'Other'	'Other'
[13] 'Not specified 4'	[8] 'Other'	'Other'
[14] 'Not specified 6'	[8] 'Other'	'Other'
[15] 'Not specified 9'	[8] 'Other'	'Other'
[16] 'Other types of ship, Category OS'	[8] 'Other'	'Other'
[17] 'Other types of ship, Category X'	[8] 'Other'	'Other'
[18] 'Other types of ship, Category Y'	[8] 'Other'	'Other'
[19] 'Other types of ship, General'	[8] 'Other'	'Other'
[20] 'Other types of ship, No additional information'	[8] 'Other'	'Other'
[21] 'Passenger ship, General'	[9] 'Passenger ship'	'Passenger ship'
[22]'Passenger ship, No additional information'	[9] 'Passenger ship'	'Passenger ship'
[23] 'Passenger ship, Reserved for future use'	[9] 'Passenger ship'	'Passenger ship'
[24] 'Pilot vessel'	[10] 'Pilot vessel'	'Harbour vessel'
[25] 'Port tender'	[12] 'Port tender'	'Harbour vessel'
[26] 'Reserved for future use, Category X'	[13] 'Reserved for future use'	'Other'
[27] 'Search and rescue vessel'	[15] 'Search and rescue vessel'	'Military vessel'
[28] 'Ship not party to an armed conflict'	[16] 'Ship not party to an armed conflict'	'Other'
[29] 'Tanker, Category OS'	[17] 'Tanker'	'Tanker'
[30] 'Tanker, Category X'	[17] 'Tanker'	'Tanker'
[31] 'Tanker, Category Y'	[17] 'Tanker'	'Tanker'
[32] 'Tanker, Category Z'	[17] 'Tanker'	'Tanker'
[33] 'Tanker, General'	[17] 'Tanker'	'Tanker'
[34] 'Tanker, No additional information'	[17] 'Tanker'	'Tanker'
[35] 'Tanker, Reserved for future use'	[17] 'Tanker'	'Tanker'
[36] 'Tug'	[19] 'Tug'	'Harbour vessel'
[37] 'Vessel, Engaged in diving operations'	[2] 'Engaged in diving operations'	'Other'
[38] 'Vessel, Engaged in dredging or underwater operations'	[3] 'Engaged in dredging or underwater operations'	'Dredging vessel'
[39] 'Vessel, Engaged in military operations'	[4] 'Engaged in military operations'	'Military vessel'
[40] 'Vessel, Fishing'	[5] 'Fishing'	'Fishing'
[41] 'Vessel, Pleasure craft'	[11] 'Pleasure craft'	'Pleasure craft'
[42] 'Vessel, Reserved for future use'	[13] 'Reserved for future use'	'Other'
[43] 'Vessel, Sailing'	[14] 'Sailing'	'Pleasure craft'
[44] 'Vessel, Towing'	[18] 'Towing'	'Harbour vessel'
[45] 'Vessel, Towing and exceeding'	[18] 'Towing'	'Harbour vessel'

TABLE 10.5: Pearson correlation coefficient matrix

	SOG	Longitude	Latitude	ROT	COG	Heading	Draught
SOG	1	-0.121738	0.151384	0.001165	0.028774	-0.144385	0.083259
Longitude	-0.121738	1	0.692813	0.042733	0.047005	0.050661	-0.297770
Latitude	0.151384	0.692813	1	0.027645	0.041401	-0.028745	-0.163515
ROT	0.001165	0.042733	0.027645	1	-0.013607	-0.026184	-0.036887
COG	0.028774	0.047005	0.041401	-0.013607	1	0.367862	-0.092212
Heading	-0.144385	0.050661	-0.028745	-0.026184	0.367862	1	-0.091885
Draught	0.083259	-0.297770	-0.163515	-0.036887	-0.092212	-0.091885	1

TABLE 10.6: Spearman's rank correlation coefficient matrix

	SOG	Longitude	Latitude	ROT	COG	Heading	Draught
SOG	1	-0.158364	0.120820	0.008972	0.098947	-0.127912	0.025607
Longitude	-0.158364	1	0.577942	0.027957	0.029294	0.031601	-0.314314
Latitude	0.120820	0.577942	1	0.039860	0.065166	-0.025763	-0.146701
ROT	0.008972	0.027957	0.039860	1	-0.003906	-0.020954	-0.025721
COG	0.098947	0.029294	0.065166	-0.003906	1	0.349988	-0.096604
Heading	-0.127912	0.031601	-0.025763	-0.020954	0.349988	1	-0.130043
Draught	0.025607	-0.314314	-0.146701	-0.025721	-0.096604	-0.130043	1

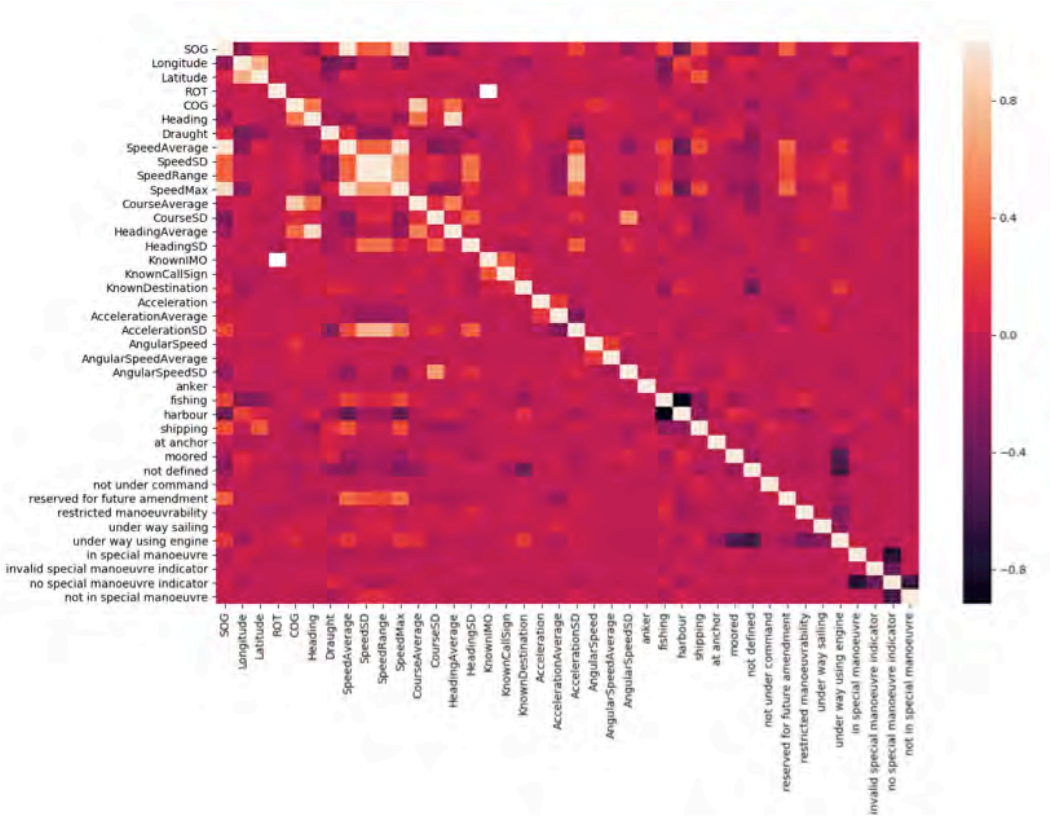

FIGURE 10.4: Visualisation of Pearson correlations

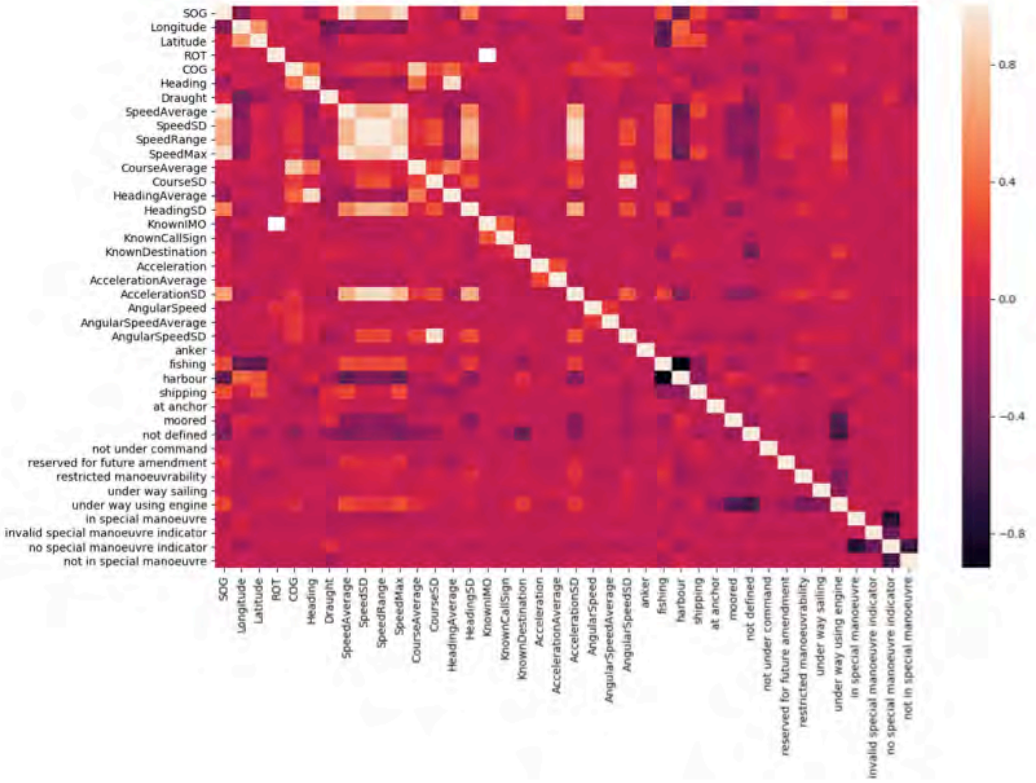

FIGURE 10.5: Visualisation of Spearman's rank correlations

TABLE 10.7: RFE random forest

Number of features	Selection of features
2	'Longitude', 'Draught'
3	'Longitude', 'Latitude', 'Draught'
4	'Longitude', 'Latitude', 'Draught', 'SpeedMax'
5	'Longitude', 'Latitude', 'Draught', 'SpeedMax', 'HeadingAverage'
6	'Longitude', 'Latitude', 'Draught', 'SpeedMax', 'HeadingAverage', 'AccelerationSD'
7	'Longitude', 'Latitude', 'Draught', 'SpeedMax', 'HeadingAverage', 'AccelerationSD', 'CourseAverage'
8	'Longitude', 'Latitude', 'Draught', 'SpeedMax', 'HeadingAverage', 'AccelerationSD', 'CourseAverage', 'HeadingSD'
9	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedMax', 'CourseAverage', 'HeadingAverage', 'AccelerationSD', 'AngularSpeedSD'
10	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedMax', 'CourseAverage', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD'
11	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedMax', 'CourseAverage', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD'
12	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedMax', 'CourseAverage', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD'
13	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined'
14	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined'
15	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined', 'restricted manoeuvrability'
16	'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined', 'restricted manoeuvrability'
17	'SOG', 'Longitude', 'Latitude', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined', 'restricted manoeuvrability'
18	'SOG', 'Longitude', 'Latitude', 'COG', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'AccelerationSD', 'AngularSpeedSD', 'not defined', 'restricted manoeuvrability'
19	'SOG', 'Longitude', 'Latitude', 'COG', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'KnownDestination', 'AccelerationSD', 'AngularSpeedSD', 'not defined', 'restricted manoeuvrability'
20	'SOG', 'Longitude', 'Latitude', 'COG', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'KnownDestination', 'AccelerationSD', 'AngularSpeedSD', 'fishing', 'not defined', 'restricted manoeuvrability'
31	'SOG', 'Longitude', 'Latitude', 'COG', 'Heading', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseAverage', 'CourseSD', 'HeadingAverage', 'HeadingSD', 'KnownCallSign', 'KnownDestination', 'AccelerationAverage', 'AccelerationSD', 'AngularSpeed', 'AngularSpeedAverage', 'AngularSpeedSD', 'fishing', 'harbour', 'at anchor', 'moored', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'no special manoeuvre indicator'

TABLE 10.8: Univariate feature selection

Number of features	Selection of features
2	'not defined', 'reserved for future amendment'
3	'not defined', 'reserved for future amendment', 'restricted manoeuvrability'
4	'SpeedRange', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability'
5	'SpeedSD', 'SpeedRange', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability'
6	'SpeedSD', 'SpeedRange', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability'
7	'SpeedSD', 'SpeedRange', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability'
8	'SpeedSD', 'SpeedRange', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing'
9	'SpeedSD', 'SpeedRange', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'not in special manoeuvre'
10	'SpeedSD', 'SpeedRange', 'SpeedMax', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'not in special manoeuvre'
11	'SpeedSD', 'SpeedRange', 'SpeedMax', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'not in special manoeuvre'
12	'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'not in special manoeuvre'
13	'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'not in special manoeuvre'
14	'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'not in special manoeuvre'
15	'SOG', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'not in special manoeuvre'
16	'SOG', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'invalid special manoeuvre indicator', 'not in special manoeuvre'
17	'SOG', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'AccelerationSD', 'fishing', 'harbour', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'invalid special manoeuvre indicator', 'not in special manoeuvre'
18	'SOG', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'HeadingSD', 'AccelerationSD', 'fishing', 'harbour', 'moored', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'invalid special manoeuvre indicator', 'not in special manoeuvre'
19	'SOG', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseSD', 'HeadingSD', 'AccelerationSD', 'fishing', 'harbour', 'moored', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'invalid special manoeuvre indicator', 'not in special manoeuvre'
20	'SOG', 'Draught', 'SpeedAverage', 'SpeedSD', 'SpeedRange', 'SpeedMax', 'CourseSD', 'HeadingSD', 'AccelerationSD', 'fishing', 'harbour', 'moored', 'not defined', 'reserved for future amendment', 'restricted manoeuvrability', 'under way sailing', 'under way using engine', 'in special manoeuvre', 'invalid special manoeuvre indicator', 'not in special manoeuvre'

TABLE 10.9: Grid search cross validation - Random forest optimisation

n_estimators	max_depth	Macro-average F1 score mean (+/-sd)	Accuracy mean (+/-sd)
50	5	0.463 (+/-0.121)	0.540 (+/-0.041)
100	5	0.468 (+/-0.117)	0.543 (+/-0.043)
200	5	0.478 (+/-0.111)	0.556 (+/-0.035)
300	5	0.470 (+/-0.105)	0.552 (+/-0.037)
400	5	0.468 (+/-0.104)	0.551 (+/-0.035)
500	5	0.471 (+/-0.105)	0.555 (+/-0.040)
50	8	0.513 (+/-0.108)	0.567 (+/-0.058)
100	8	0.531 (+/-0.046)	0.568 (+/-0.048)
200	8	0.533 (+/-0.046)	0.570 (+/-0.046)
300	8	0.533 (+/-0.043)	0.571 (+/-0.042)
400	8	0.518 (+/-0.087)	0.573 (+/-0.042)
500	8	0.518 (+/-0.090)	0.572 (+/-0.040)
50	10	0.552 (+/-0.076)	0.577 (+/-0.072)
100	10	0.557 (+/-0.077)	0.583 (+/-0.072)
200	10	0.557 (+/-0.078)	0.582 (+/-0.074)
300	10	0.559 (+/-0.077)	0.581 (+/-0.073)
400	10	0.554 (+/-0.082)	0.575 (+/-0.081)
500	10	0.550 (+/-0.077)	0.575 (+/-0.079)
50	20	0.594 (+/-0.037)	0.579 (+/-0.057)
100	20	0.601 (+/-0.065)	0.589 (+/-0.078)
200	20	0.599 (+/-0.061)	0.588 (+/-0.078)
300	20	0.602 (+/-0.064)	0.591 (+/-0.087)
400	20	0.602 (+/-0.059)	0.591 (+/-0.078)
500	20	0.604 (+/-0.058)	0.592 (+/-0.078)
50	None	0.587 (+/-0.078)	0.572 (+/-0.084)
100	None	0.595 (+/-0.072)	0.583 (+/-0.082)
200	None	0.599 (+/-0.060)	0.588 (+/-0.065)
300	None	0.602 (+/-0.061)	0.589 (+/-0.065)
400	None	0.599 (+/-0.059)	0.589 (+/-0.064)
500	None	0.600 (+/-0.056)	0.588 (+/-0.069)

TABLE 10.10: Confusion matrix random forest

		Predicted								
		Cargo	Dredging	Fishing	Harbour	Military	Other	Passenger	Pleasure	Tanker
True	Cargo	4266	334	0	497	0	23	403	0	653
	Dredging	534	110	0	53	38	24	0	0	96
	Fishing	301	0	0	0	0	0	0	0	0
	Harbour	234	17	0	1467	0	22	67	0	0
	Military	1	0	0	2	174	89	44	0	127
	Other	793	12	0	85	0	97	0	0	0
	Passenger	354	0	0	527	0	0	45	0	0
	Pleasure	11	0	0	0	1	0	0	0	0
	Tanker	545	4	0	32	0	8	0	0	297

TABLE 10.11: Grid search cross validation - Support vector machine optimisation

C	gamma	Macro-average F1 score mean (+/-sd)	Accuracy mean (+/-sd)
10	1e-09	0.055 (+/-0.000)	0.379 (+/-0.000)
10	1e-07	0.055 (+/-0.000)	0.379 (+/-0.000)
10	1e-05	0.098 (+/-0.064)	0.416 (+/-0.061)
10	0.001	0.398 (+/-0.136)	0.499 (+/-0.063)
10	0.1	0.468 (+/-0.103)	0.455 (+/-0.072)
100	1e-09	0.055 (+/-0.000)	0.379 (+/-0.000)
100	1e-07	0.060 (+/-0.009)	0.382 (+/-0.007)
100	1e-05	0.324 (+/-0.060)	0.488 (+/-0.062)
100	0.001	0.433 (+/-0.171)	0.514 (+/-0.084)
100	0.1	0.493 (+/-0.107)	0.445 (+/-0.124)
1000	1e-09	0.055 (+/-0.000)	0.379 (+/-0.000)
1000	1e-07	0.098 (+/-0.064)	0.416 (+/-0.061)
1000	1e-05	0.390 (+/-0.128)	0.486 (+/-0.051)
1000	0.001	0.426 (+/-0.133)	0.503 (+/-0.045)
1000	0.1	0.503 (+/-0.061)	0.466 (+/-0.066)
10000	1e-09	0.059 (+/-0.010)	0.382 (+/-0.008)
10000	1e-07	0.333 (+/-0.062)	0.481 (+/-0.067)
10000	1e-05	0.407 (+/-0.093)	0.500 (+/-0.042)
10000	0.001	0.467 (+/-0.072)	0.480 (+/-0.060)
10000	0.1	0.479 (+/-0.012)	0.439 (+/-0.066)
100000	1e-09	0.148 (+/-0.112)	0.326 (+/-0.234)
100000	1e-07	0.412 (+/-0.102)	0.512 (+/-0.041)
100000	1e-05	0.425 (+/-0.133)	0.508 (+/-0.084)
100000	0.001	0.500 (+/-0.107)	0.497 (+/-0.131)
100000	0.1	0.489 (+/-0.049)	0.438 (+/-0.058)

TABLE 10.12: Confusion matrix support vector machine

		Predicted								
		Cargo	Dredging	Fishing	Harbour	Military	Other	Passenger	Pleasure	Tanker
True	Cargo	5416	241	0	84	0	0	1065	0	0
	Dredging	611	244	0	0	0	0	0	0	0
	Fishing	301	0	0	0	0	0	0	0	0
	Harbour	528	349	0	833	0	97	0	0	0
	Military	134	0	0	0	303	0	0	0	0
	Other	890	1	0	93	0	3	0	0	0
	Passenger	40	0	0	102	315	0	469	0	0
	Pleasure	12	0	0	0	0	0	0	0	0
	Tanker	817	0	0	0	0	0	0	0	69

TABLE 10.13: Grid search cross validation - Artificial neural network optimisation

max_iter	hidden_layer_sizes	activation	alpha	Macro-average F1 score mean (+/-sd)	Accuracy mean (+/-sd)
100	4	'logistic'	1e-05	0.283 (+/-0.058)	0.439 (+/-0.140)
500	4	'logistic'	1e-05	0.280 (+/-0.049)	0.440 (+/-0.127)
100	6	'logistic'	1e-05	0.236 (+/-0.169)	0.419 (+/-0.156)
500	6	'logistic'	1e-05	0.284 (+/-0.157)	0.451 (+/-0.129)
100	7	'logistic'	1e-05	0.287 (+/-0.036)	0.421 (+/-0.142)
500	7	'logistic'	1e-05	0.311 (+/-0.129)	0.428 (+/-0.132)
100	8	'logistic'	1e-05	0.245 (+/-0.092)	0.424 (+/-0.069)
500	8	'logistic'	1e-05	0.273 (+/-0.130)	0.429 (+/-0.057)
100	9	'logistic'	1e-05	0.316 (+/-0.042)	0.418 (+/-0.129)
500	9	'logistic'	1e-05	0.320 (+/-0.043)	0.418 (+/-0.116)
100	10	'logistic'	1e-05	0.337 (+/-0.147)	0.386 (+/-0.182)
500	10	'logistic'	1e-05	0.330 (+/-0.128)	0.382 (+/-0.173)
100	11	'logistic'	1e-05	0.350 (+/-0.071)	0.438 (+/-0.136)
500	11	'logistic'	1e-05	0.334 (+/-0.056)	0.422 (+/-0.132)
100	4	'logistic'	0.001	0.282 (+/-0.057)	0.439 (+/-0.138)
500	4	'logistic'	0.001	0.280 (+/-0.050)	0.440 (+/-0.127)
100	6	'logistic'	0.001	0.256 (+/-0.220)	0.430 (+/-0.180)
500	6	'logistic'	0.001	0.278 (+/-0.178)	0.450 (+/-0.107)
100	7	'logistic'	0.001	0.282 (+/-0.043)	0.418 (+/-0.141)
500	7	'logistic'	0.001	0.318 (+/-0.113)	0.425 (+/-0.146)
100	8	'logistic'	0.001	0.271 (+/-0.089)	0.426 (+/-0.069)
500	8	'logistic'	0.001	0.281 (+/-0.078)	0.421 (+/-0.060)
100	9	'logistic'	0.001	0.326 (+/-0.054)	0.418 (+/-0.125)
500	9	'logistic'	0.001	0.337 (+/-0.070)	0.431 (+/-0.120)
100	10	'logistic'	0.001	0.334 (+/-0.139)	0.386 (+/-0.175)
500	10	'logistic'	0.001	0.338 (+/-0.106)	0.394 (+/-0.164)
100	11	'logistic'	0.001	0.349 (+/-0.059)	0.443 (+/-0.127)
500	11	'logistic'	0.001	0.349 (+/-0.036)	0.444 (+/-0.107)
100	4	'logistic'	0.1	0.270 (+/-0.113)	0.433 (+/-0.142)
500	4	'logistic'	0.1	0.271 (+/-0.109)	0.431 (+/-0.144)
100	6	'logistic'	0.1	0.271 (+/-0.169)	0.445 (+/-0.129)
500	6	'logistic'	0.1	0.268 (+/-0.168)	0.440 (+/-0.161)
100	7	'logistic'	0.1	0.302 (+/-0.122)	0.421 (+/-0.150)
500	7	'logistic'	0.1	0.300 (+/-0.112)	0.421 (+/-0.147)
100	8	'logistic'	0.1	0.247 (+/-0.111)	0.415 (+/-0.127)
500	8	'logistic'	0.1	0.246 (+/-0.117)	0.407 (+/-0.162)
100	9	'logistic'	0.1	0.300 (+/-0.055)	0.402 (+/-0.117)
500	9	'logistic'	0.1	0.288 (+/-0.063)	0.398 (+/-0.123)
100	10	'logistic'	0.1	0.323 (+/-0.078)	0.422 (+/-0.148)
500	10	'logistic'	0.1	0.325 (+/-0.085)	0.420 (+/-0.146)
100	11	'logistic'	0.1	0.338 (+/-0.136)	0.437 (+/-0.159)
500	11	'logistic'	0.1	0.332 (+/-0.094)	0.430 (+/-0.136)
100	4	'tanh'	1e-05	0.249 (+/-0.112)	0.410 (+/-0.149)
500	4	'tanh'	1e-05	0.242 (+/-0.122)	0.426 (+/-0.126)
100	6	'tanh'	1e-05	0.124 (+/-0.098)	0.396 (+/-0.080)
500	6	'tanh'	1e-05	0.124 (+/-0.097)	0.397 (+/-0.079)
100	7	'tanh'	1e-05	0.191 (+/-0.161)	0.426 (+/-0.107)
500	7	'tanh'	1e-05	0.230 (+/-0.263)	0.441 (+/-0.119)
100	8	'tanh'	1e-05	0.243 (+/-0.098)	0.398 (+/-0.068)
500	8	'tanh'	1e-05	0.236 (+/-0.095)	0.392 (+/-0.059)
100	9	'tanh'	1e-05	0.238 (+/-0.117)	0.418 (+/-0.110)
500	9	'tanh'	1e-05	0.252 (+/-0.154)	0.422 (+/-0.120)
100	10	'tanh'	1e-05	0.207 (+/-0.153)	0.374 (+/-0.070)
500	10	'tanh'	1e-05	0.246 (+/-0.082)	0.398 (+/-0.127)
100	11	'tanh'	1e-05	0.336 (+/-0.087)	0.429 (+/-0.172)
500	11	'tanh'	1e-05	0.332 (+/-0.079)	0.430 (+/-0.173)
100	4	'tanh'	0.001	0.249 (+/-0.112)	0.410 (+/-0.149)
500	4	'tanh'	0.001	0.259 (+/-0.106)	0.413 (+/-0.142)
100	6	'tanh'	0.001	0.198 (+/-0.142)	0.411 (+/-0.108)
500	6	'tanh'	0.001	0.191 (+/-0.141)	0.422 (+/-0.086)
100	7	'tanh'	0.001	0.258 (+/-0.232)	0.429 (+/-0.118)
500	7	'tanh'	0.001	0.261 (+/-0.245)	0.432 (+/-0.132)
100	8	'tanh'	0.001	0.290 (+/-0.074)	0.423 (+/-0.041)
500	8	'tanh'	0.001	0.312 (+/-0.134)	0.420 (+/-0.055)
100	9	'tanh'	0.001	0.297 (+/-0.086)	0.433 (+/-0.132)
500	9	'tanh'	0.001	0.298 (+/-0.106)	0.436 (+/-0.127)

TABLE 10.14: Grid search cross validation - Artificial neural network optimisation (continued)

max_iter	hidden_layer_sizes	activation	alpha	Macro-average F1 score mean (+/-sd)	Accuracy mean (+/-sd)
100	10	'tanh'	0.001	0.328 (+/-0.117)	0.452 (+/-0.061)
500	10	'tanh'	0.001	0.323 (+/-0.110)	0.452 (+/-0.064)
100	11	'tanh'	0.001	0.344 (+/-0.135)	0.399 (+/-0.174)
500	11	'tanh'	0.001	0.348 (+/-0.142)	0.406 (+/-0.185)
100	4	'tanh'	0.1	0.249 (+/-0.101)	0.406 (+/-0.137)
500	4	'tanh'	0.1	0.247 (+/-0.104)	0.421 (+/-0.123)
100	6	'tanh'	0.1	0.281 (+/-0.088)	0.437 (+/-0.058)
500	6	'tanh'	0.1	0.280 (+/-0.088)	0.434 (+/-0.067)
100	7	'tanh'	0.1	0.247 (+/-0.185)	0.425 (+/-0.147)
500	7	'tanh'	0.1	0.247 (+/-0.185)	0.425 (+/-0.147)
100	8	'tanh'	0.1	0.269 (+/-0.087)	0.418 (+/-0.108)
500	8	'tanh'	0.1	0.269 (+/-0.087)	0.418 (+/-0.108)
100	9	'tanh'	0.1	0.278 (+/-0.083)	0.416 (+/-0.111)
500	9	'tanh'	0.1	0.278 (+/-0.083)	0.416 (+/-0.111)
100	10	'tanh'	0.1	0.293 (+/-0.067)	0.419 (+/-0.179)
500	10	'tanh'	0.1	0.293 (+/-0.071)	0.415 (+/-0.180)
100	11	'tanh'	0.1	0.347 (+/-0.024)	0.434 (+/-0.135)
500	11	'tanh'	0.1	0.350 (+/-0.028)	0.438 (+/-0.139)
100	4	'relu'	1e-05	0.320 (+/-0.133)	0.431 (+/-0.129)
500	4	'relu'	1e-05	0.317 (+/-0.120)	0.429 (+/-0.128)
100	6	'relu'	1e-05	0.294 (+/-0.132)	0.453 (+/-0.100)
500	6	'relu'	1e-05	0.299 (+/-0.073)	0.434 (+/-0.117)
100	7	'relu'	1e-05	0.277 (+/-0.065)	0.431 (+/-0.064)
500	7	'relu'	1e-05	0.277 (+/-0.065)	0.431 (+/-0.064)
100	8	'relu'	1e-05	0.290 (+/-0.119)	0.444 (+/-0.087)
500	8	'relu'	1e-05	0.276 (+/-0.087)	0.445 (+/-0.088)
100	9	'relu'	1e-05	0.313 (+/-0.052)	0.430 (+/-0.148)
500	9	'relu'	1e-05	0.313 (+/-0.052)	0.430 (+/-0.148)
100	10	'relu'	1e-05	0.302 (+/-0.187)	0.450 (+/-0.136)
500	10	'relu'	1e-05	0.300 (+/-0.185)	0.447 (+/-0.133)
100	11	'relu'	1e-05	0.363 (+/-0.042)	0.438 (+/-0.074)
500	11	'relu'	1e-05	0.402 (+/-0.044)	0.442 (+/-0.110)
100	4	'relu'	0.001	0.320 (+/-0.133)	0.436 (+/-0.119)
500	4	'relu'	0.001	0.316 (+/-0.121)	0.435 (+/-0.118)
100	6	'relu'	0.001	0.294 (+/-0.112)	0.433 (+/-0.080)
500	6	'relu'	0.001	0.292 (+/-0.111)	0.431 (+/-0.077)
100	7	'relu'	0.001	0.321 (+/-0.128)	0.443 (+/-0.069)
500	7	'relu'	0.001	0.313 (+/-0.104)	0.438 (+/-0.064)
100	8	'relu'	0.001	0.292 (+/-0.118)	0.445 (+/-0.087)
500	8	'relu'	0.001	0.278 (+/-0.086)	0.446 (+/-0.088)
100	9	'relu'	0.001	0.313 (+/-0.052)	0.430 (+/-0.148)
500	9	'relu'	0.001	0.313 (+/-0.052)	0.430 (+/-0.148)
100	10	'relu'	0.001	0.313 (+/-0.154)	0.435 (+/-0.136)
500	10	'relu'	0.001	0.313 (+/-0.154)	0.435 (+/-0.136)
100	11	'relu'	0.001	0.377 (+/-0.089)	0.435 (+/-0.061)
500	11	'relu'	0.001	0.387 (+/-0.079)	0.445 (+/-0.065)
100	4	'relu'	0.1	0.332 (+/-0.107)	0.434 (+/-0.133)
500	4	'relu'	0.1	0.329 (+/-0.096)	0.430 (+/-0.127)
100	6	'relu'	0.1	0.274 (+/-0.090)	0.436 (+/-0.093)
500	6	'relu'	0.1	0.296 (+/-0.133)	0.442 (+/-0.106)
100	7	'relu'	0.1	0.275 (+/-0.016)	0.426 (+/-0.096)
500	7	'relu'	0.1	0.275 (+/-0.016)	0.426 (+/-0.096)
100	8	'relu'	0.1	0.283 (+/-0.134)	0.444 (+/-0.099)
500	8	'relu'	0.1	0.283 (+/-0.134)	0.444 (+/-0.099)
100	9	'relu'	0.1	0.303 (+/-0.042)	0.455 (+/-0.093)
500	9	'relu'	0.1	0.303 (+/-0.042)	0.455 (+/-0.093)
100	10	'relu'	0.1	0.302 (+/-0.173)	0.446 (+/-0.135)
500	10	'relu'	0.1	0.302 (+/-0.173)	0.446 (+/-0.135)
100	11	'relu'	0.1	0.358 (+/-0.043)	0.424 (+/-0.105)
500	11	'relu'	0.1	0.358 (+/-0.043)	0.424 (+/-0.105)

TABLE 10.15: Confusion matrix artificial neural network

		Predicted								
		Cargo	Dredging	Fishing	Harbour	Military	Other	Passenger	Pleasure	Tanker
True	Cargo	4818	50	0	1433	0	1	0	263	241
	Dredging	608	0	0	86	0	151	0	10	0
	Fishing	299	0	0	0	0	0	0	0	2
	Harbour	1331	0	0	439	0	0	0	14	23
	Military	23	0	0	153	0	172	0	89	0
	Other	722	0	0	97	0	97	0	71	0
	Passenger	325	0	0	601	0	0	0	0	0
	Pleasure	8	0	0	2	0	0	0	2	0
	Tanker	879	0	0	0	0	0	0	3	4

TABLE 10.16: List of ideas for feature engineering

Feature name	Description
Bearing	Angle in relation to the destination (between 0 and 359 degrees)
Appropriate behaviour	E.g., whether a vessel is sailing in the right direction if the location is a shipping lane. Or whether the location of a vessel is not changing if it is located in an anchor area.
Course change rate	Percentage of time that the vessel has spent changing course at a rate of between 5 and 10 degrees per minute. (source: Mascaro et al. (2010) [20])
Heading change rate	Percentage of time that the vessel has spent changing heading at a rate of between 5 and 10 degrees per minute.
Ship size	Combination of the vessels length, width and draught ($length \cdot width \cdot draught$). (source: Mascaro et al. (2010) [20])
Straight	Number of times the vessel has travelled straight (defined as the course changing by less than 1 degree per second when not stopped). (source: Mascaro et al. (2010) [20])
Average recorded speed	The average speed recorded for the vessel by marinetraffic.com or digital-seas. (source: Mascaro et al. (2010) [20])
Maximum recorded speed	The maximum speed recorded for the vessel by marinetraffic.com or digital-seas.
Number close interactions	Number of other vessels close by. (source: Mascaro et al. (2010) [20])
Closest type	The type of the closest vessel. (source: Mascaro et al. (2010) [20])

References

[1] *AIS messages*. 2017. URL: https://www.navcen.uscg.gov/?pageName=AISMessages.

[2] Scikit-learn developers (BSD License). *MLP Classifier*. 2018. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier (visited on 12/11/2018).

[3] Scikit-learn developers (BSD License). *Neural network models (supervised) - Mathematical formulation*. 2018. URL: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#mathematical-formulation (visited on 12/11/2018).

[4] Scikit-learn developers (BSD License). *Random Forest Classifier*. 2018. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 12/11/2018).

[5] Scikit-learn developers (BSD License). *Support Vector Classification*. 2018. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC (visited on 12/11/2018).

[6] Scikit-learn developers (BSD License). *Support Vector Machines*. 2018. URL: https://scikit-learn.org/stable/modules/svm.html (visited on 12/11/2018).

[7] Theodoros Evgeniou and Massimiliano Pontil. "Support Vector Machines: Theory and Applications". In: *Machine Learning and Its Applications: Advanced Lectures*. Vol. 2049. Jan. 2001, pp. 249–257. DOI: 10.1007/3-540-44673-7_12.

[8] Rafael Falcon, Rami Abielmona, and Erik Blasch. "Behavioral Learning of Vessel Types with Fuzzy-Rough Decision Trees". In: *FUSION 2014 - 17th International Conference on Information Fusion*. July 2014.

[9] Tristan Fletcher. "Support Vector Machines Explained". Jan. 2009.

[10] Venkat Gudevada, Amy Apon, and Junhua Ding. "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations". In: *International Journal on Advances in Software* 10 (July 2017), pp. 1–20.

[11] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46.1 (Jan. 2002), pp. 389–422. DOI: 10.1023/A:1012487302797. URL: https://doi.org/10.1023/A:1012487302797.

[12] Jiawei Han, Micheline Kamber, and Jian Pei. "Data Mining. Concepts and Techniques, 3rd Edition". In: *Waltham: Elsevier* (2012).

[13] Douglas Hawkins. "The Problem of Overfitting". In: *Journal of chemical information and computer sciences* 44 (May 2004), pp. 1–12. DOI: 10.1021/ci0342472.

[14] *How AIS works*. 2016. URL: https://www.navcen.uscg.gov/?pageName=AISworks.

[15] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. *A Practical Guide to Support Vector Classification*. 2003. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[16] IBM. *CART Algorithm*. URL: ftp://ftp.boulder.ibm.com/software/analytics/spss/support/Stats/Docs/Statistics/Algorithms/14.0/TREE-CART.pdf.

[17] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations* (Dec. 2014). URL: https://arxiv.org/pdf/1412.6980.pdf.

[18] Rikard Laxhammar. "Anomaly detection for sea surveillance". In: *11th International Conference on Information Fusion*. June 2008, pp. 1–8.

[19] Henrik Ljunggren. "Exploring the capabilities of deep learning in sea surveillance". MA thesis. School of Industrial Engineering and Management, June 2017.

[20] Steven Mascaro, Kevin B Korb, and Ann E Nicholson. "Learning Abnormal Vessel Behaviour from AIS Data with Bayesian Networks at Two Time Scales". In: *Tracks a Journal of Artists Writings* (2010), pp. 1–34.

[21] Stefano Nembrini, Inke R König, and Marvin N Wright. "The revival of the Gini importance?" In: *Bioinformatics* 34.21 (May 2018), pp. 3711–3718. DOI: 10.1093/bioinformatics/bty373. eprint: http://oup.prod.sis.lan/bioinformatics/article-pdf/34/21/3711/26146978/bty373.pdf. URL: https://dx.doi.org/10.1093/bioinformatics/bty373.

[22] Giuliana Pallotta, Michele Vespe, and Karna Bryan. "Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction". In: *Entropy* 15 (June 2013), pp. 2218–2245. DOI: 10.3390/e15062218.

[23] Foram S. Panchal and Mahesh Panchal. "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network". In: *International Journal of Computer Science and Mobile Computing* 3.11 (Nov. 2014), pp. 455–464. URL: https://www.ijcsmc.com/docs/papers/November2014/V3I11201499a19.pdf.

[24] Alexandros Sfyridis, T Cheng, and Michele Vespe. "Detecting Vessels Carrying Migrants using Machine Learning". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. IV-4/W2. Oct. 2017, pp. 53–60. DOI: 10.5194/isprs-annals-IV-4-W2-53-2017.

[25] *TACTICOS - Worlds favourite Combat Management System*. Thales Nederland B.V. 2018.

[26] Mauricio Roberto Veronez et al. "Regional Mapping of the Geoid Using GNSS (GPS) Measurements and an Artificial Neural Network". In: *Remote Sensing* 3.4 (2011), pp. 668–683. DOI: 10.3390/rs3040668. URL: http://www.mdpi.com/2072-4292/3/4/668.

[27] *What variables are important in predicting bovine viral diarrhea virus? A random forest approach - Scientific Figure on ResearchGate*. URL: https://www.researchgate.net/figure/Random-forest-model-Example-of-training-and-classification-processes-using-random_fig5_280533599 (visited on 01/02/2019).

[28] *Wie zijn wij? | Thales Group*. 2018. URL: https://www.thalesgroup.com/en/worldwide/careers/wie-zijn-wij.

[29] Ian Witten and Eibe Frank. *Data Mining - Practical Machine Learning Tools and Techniques*. Second Edition. Morgan Kaufmann - Elsevier, June 2005.

[30] *Workday and Thales Group*. 2019. URL: https://www.workday.com/en-hk/customers/thales.html.

[31] Brian L. Young. *Predicting vessel trajectories from AIS data using R*. June 2017. URL: https://calhoun.nps.edu/handle/10945/55564.