MASTER'S THESIS 2023

Investigating the Applicability of Deep Learning to Profile Ship Risk

Mathias Kindberg

ISSN 1650-2884
LU-CS-EX: 2023-11

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2023-11

Investigating the Applicability of Deep Learning to Profile Ship Risk

Undersökning av applicerbarheten för maskinintelligens till att profilera fartygsrisk

Mathias Kindberg

Investigating the Applicability of Deep Learning to Profile Ship Risk

Mathias Kindberg
mathias.kindberg@gmail.com

April 27, 2023

Master's thesis work carried out at RISE RESEARCH INSTITUTES OF SWEDEN.

Supervisors: Pierre Nugues, pierre.nugues@cs.lth.se
Johannes Hüffmeier, johannes.huffmeier@ri.se,
Luis Sanchez-Heres, luis.sanchez-heres@ri.se

Examiner: Elin A. Topp, elin_anna.topp@cs.lth.se

Abstract

This thesis investigates the applicability of machine learning and deep learning models to increase ship safety through predicting if an individual ship will pass the next port state control, and then how many deficiencies should be encountered. Previous studies have shown a strong link between vessel safety and the outcome of the ship's next port state control inspection. Starting with a literature review of the current state of the art, we bring the statistical conclusions and correlations down to the level of predicting the inspection outcome for a single ship.

As input, we used static and dynamic ship data from the maritime industry. We collected a tabular dataset covering all the port state control protocols from 2016 to 2020 related to Paris Memorandum of Understanding, ship data from Clarksons Research, IHS databases, and MRV (Monitoring, Reporting and Validation) EU emission data.

We then applied random forests and deep learning on this dataset. As a result, we could predict if the next port state control will be a detention or not with a 72% accuracy and a F-Measure of 0.704. We also ranked the feature importance and found new measures not mentioned in previous research.

As a conclusion, we find the same statistical signal in regards to vessel type, etc., but have a hard time creating a future port state control prediction model, which is accurate enough on a sample basis. The thesis finishes with lessons learned from working on this problem and this dataset. We finally hope that our research will find applications and lead to further research.

Keywords: maritime safety, ship profiling, deep learning, machine learning, port state control, neural network

Acknowledgements

I would like to to thank my supervisors at RISE, Research Institutes of Sweden, Johannes Hüffmeier, and Luis Sanchez-Heres, for their ongoing support throughout the whole project. From helping to find data sources, considering algorithms, feedback on decisions to the final very extensive reviewing of the report. This thesis would not have been possible without their help.

I would also like to thank Viktor Norrsjö at RISE for his help when problems arose regarding the implementation of specific details in the machine learning algorithms and the datasets I was working with.

Finally a big thank you to my supervisor at LTH, Pierre Nugues, helping me shape this thesis to its final form.

Contents

Abbreviations used

- AIS - Automatic Identifcation System
- BTH - Blekinge Tekniska Högskola
- CLIA - Cruise Lines International Assoication
- DAD - Deep Anomaly Detection
- EEA - European Economic Area
- EU - European Union
- FCN - Fully Convolutional Network
- GAN - Generative Adversarial Networks
- GAP - Global Average Pooling
- GT - Gross Tonnage (Volume metric)
- HELCOM - Helsinki Commission
- IACS - International Association of Classification Societies
- ILO - International Labour Organization
- IMO - International Maritime Organization
- ISM - International Safety Management
- JSON - JavaScript Object Notation (Data format)
- LSTM - Long Short-Term Memory
- MAE - Mean Absolute Error
- MMSI - Maritime Mobile Service Identiy
- MoU - Memorandum of Understanding
- MRV - Monitoring Reporting Validation
- PSC - Port State Control
- RMSE - Real Mean Squared Error
- RO - Recognized Organization
- Ro-Ro - Roll on - Roll off
- SMC - Safe Manning Certificate

- SOLAS - Safety Of Lives At Sea
- SVM - Support Vector Machine
- UNCTAD - United Nations Conference on Trade and Development
- USCG - United States Coast Guard
- VLCC - Very Large Crude Carrier
- VTS - Vessel traffic Services

Chapter 1

Introduction

Maritime safety describes the shipping sector's work to minimize risk for incidents and accidents leading, in the worst cases, to loss of life, environmental damage, and loss of property. It covers everything from ship construction and land organization to the maintenance of the ship, extending to the crew onboard and their education, training, and professionalism. Existing methods, indices, and quantification of single vessel risk profiles consist of a simple scoring system based on statistically gathered data. In this thesis, we connect these statistical findings with the individual vessel data and explore the possibilities of predicting future performance to advance the current knowledge of maritime safety.

In this first section, we aim to give the reader a brief background on the maritime industry and the relevance of the problem formulation.

1.1 Short maritime background

This chapter gives the reader a short introduction to the shipping industry and its data usage, describing the specifics of the industry for the problem at hand.

1.1.1 Shipping in General

Shipping is a global industry with a long history and is the primary mode of transporting goods and partly passenger travel. It is an international, non-standardized business with different rules for national and international traffic, leading to a very heterogeneous problem space. There is also no common sourcing of information except the most basic characteristics. Instead, commercial providers exist to fill this need.

1.1.2 Maritime Safety

Maritime safety has improved from historically being a risky industry for the seaman to one of the safest means of transportation. Rules and regulations have been traditionally developed based on severe accidents such as those of Titanic, Herald of Free Enterprise, Estonia, and Prestige.

Research on maritime safety and safety in general shows that proactive approaches have the potential to improve safety (Hollnagel, 2018). Newer regulations require a proactive approach toward safety and have started to be adopted by ship owners and shipping companies. Cargo owners are partly enforcing them through, for example, the vetting regime for tankers.

The safety level on ships still differs widely, and many factors steer the safety performance of a specific ship. It is therefore challenging to identify risk levels based on a single ship compared to a grouping. Earlier studies (Li et al., 2014b; Hänninen and Kujala, 2014) have shown statistical correlations between specific features of a ship and the probability of the ship's involvement in accidents.

1.1.3 Machine Learning in the Maritime Industry

When doing a literature review of data applications in the maritime industry, we found that machine learning has been used very sparingly when referring to incidents and inspections, with no articles detailing deep learning. Most other studies center around positional data and other easy-to-get public data and, from some angle, infers based on this. In Section 2.2, we review the literature of this field.

1.2 Relevance

Safety is paramount in the maritime industry. A single mishap may have significant consequences, both environmentally and by causing debilitating injuries among the crews. A central theme is to increase overall safety with the limited resources available. One method is to vary the inspection rate and, thus, through correct profiling to see individual vessel differences. Another issue in the maritime industry is the frequent blaming of the crew instead of instituting structural changes.

Previous work in the field centers on statistical studies and has found statistically significant connections between vessel data, future inspections, and incidents. The thesis's relevance is whether these statistical results can be applied on a single vessel basis using deep learning, two fields that have not been connected in earlier research.

1.3 Goal

This thesis investigates the possibility of classifying vessel risk using publicly available data. The goal is to expand on previous work in the field with a larger, more diverse dataset and more advanced models taking advantage of modern breakthroughs in machine learning. We will construct a baseline classifier with a known, generally applicable method based on the

available data. After that, we will construct a deep learning model trying to improve on the baseline.

1.4 Research Questions

Based on the general goal of the thesis, we break it down into three research questions:

RQ1 The validity and performance of a baseline model to predict a ship's future state regarding its safe operations.

RQ2 The possibility of using deep learning models to predict a ship's future state regarding its safe operations and performance.

RQ3 How do the results of the constructed models relate to previous statistical research? When are the conclusions similar, and most importantly, when do they diverge?

1.5 Methodology

The methodology of the thesis consists of the following steps:

- Literature review
- Collection of data
- Identification and comparison of machine learning models based on the data
- Evaluation

Chapter 2 describes the literature review used to construct the theoretical basis; Chapter 3 gives insights into how data was collected and prepared. Chapter 4 gives details on the models implemented and their results. The thesis concludes with a discussion of the results in Chapter 5 followed by conclusions in Chapter 6.

1.6 Limitations

Due to the maritime industry having many facets, a subset of all data available needs to be selected. Possible data sources are investigated in Chapters 2 and 3. An example of the many facets is comparing a tiny passenger vessel running tours in a canal which is under a completely different rule regime compared to a 400m container ship. Then another completely different rule regime is applied when comparing an oil tanker to the ship types mentioned above.

To select ships with more harmonized rules, we developed a selection mechanism based on the current rules applied to commercial ships. We designed the criteria to be inclusive rather than exclusionary and let the Port State Control (PSC) inspections be the primary selection mechanism. This choice may introduce bias by the inspectors but allows us to access the

most accurate information regarding the largest number of vessels. We explain the statistical correlations found by earlier research in Chapter 2.

The criteria previously mentioned for the selection of ships are based on the following mechanisms.

- Gross Tonnage equal to or larger than 300. Which is the Safety of Lives At Sea (SOLAS) limit for cargo ships on international voyages (100 GT for passenger ships.)
- Registered MMSI number (Maritime Mobile Service Identity). A globally unique number identifying the radio installation.
- No fishing vessels. They are not included in the PSC regime as fishing vessels have a particular, often high-risk operational profile.
- Only readily available data sources were used with the specific limitations given by each data set regarding sourcing, errors, and sources of uncertainties.

Chapter 2
Theory and Background

In this thesis, we bring together two areas that we believe have not been connected in previous research. One side is the data-driven analyses of maritime safety-related data. Earlier research in this field starts from a statistics side, continually evolving with more advanced models as the field expands. The other side is deep learning. The application here is a crosscut over several different sub-fields. The data is primarily categorical with a temporal component. The task at hand can also be seen as anomaly detection since an unsafe vessel is an anomaly in an otherwise safe system.

The structure of this chapter is to first give an overview of the maritime industry in Section 2.1, which we then follow with a literature review of data-driven analyses of maritime data in Section 2.2. To give an overview of another applicable risk index, we describe the Paris MoU Risk Index in Section 2.4. The machine and deep learning sides are explored in Chapter 2.5. Finally, for validating the models we will be creating, we describe the applicable methods in Section 2.6.

2.1 Background

In this section, the reader is given a brief overview of data, machine learning, and the maritime industry, together with a brief history and goals of the datasets used in the developed models. For the literature review regarding data and the maritime industry, see Section 2.2. We detail the gathering, selection, and cleaning of data sources in Chapter 3.

2.1.1 Summary of Shipping

The maritime sector is a global industry with a long history, spanning everything from canal tours to container freight to the offshore industry, only to name a few areas. In 2019, according to the United Nations Conference on Trade and Development (UNCTAD, 2019), the maritime sector transported goods weighing more than 11.09 billion tons, moved by a fleet

with a total tonnage of 1.98 billion DWT (dead-weight tonnage, measure of the total cargo and loading capacity of a ship).

On the passenger side, the cruise industry carried 28.5 million passengers in 2018, as given by the Cruise Lines International Association (2019). It is tough to put an accurate number on the scale of the passenger transport side. This part of the industry ranges from local public transportation to longer routes. For example, the association for coastal traffic in Sweden, which organizes 110 companies comprising 330 vessels, most of which are too small to be included in this thesis, transports over 35 million passengers and 12 million vehicles yearly[1]. An example of larger ferries on longer routes is Stena Line, having many routes connecting Europe, transporting 7.5 million passengers, 1.7 million vehicles, and 2.1 million cargo units yearly[2].

The Swedish part mentioned above is a speck of an enormous global industry. In monetary value, the seaborne trade carries around 70% of the total global trade. The value of this trade is hard to estimate, but UNCTAD suggests 14 trillion US dollars per year. The industry is growing by 2-3% every year (UNCTAD, 2019).

2.1.2 Port State Control

Port State Control is a global system with distinct regional implementations. The overarching mission is to eliminate the operation of sub-standard ships through a harmonized system. Systems exist outside of the regionalized harmonized system. The United States, for example, operates a separate but similar system through the United States Coast Guard (USCG). The port state control regions are shown in Figure 2.1.

The port state control scheme was created in 1978 by fourteen countries in western Europe, mainly dealing with shipboard living and working conditions as described by the International Labour Organization (ILO) convention no. 147. However, right when it was about to come into effect, a massive oil spill happened off the coast of France: the grounding of the Very Large Crude Carrier (VLCC) *Amoco Cadiz*[3]. As a result, a more comprehensive memorandum was created, broadening the focus also to include the safety of lives at sea and the prevention of pollution. This memorandum has later been amended several times, and many new member countries have been added. The organization today consists of 27 participating countries and covers the waters of the European coastal states and the North Atlantic basin from North America to Europe.

The basic principle is that the prime responsibility for compliance with the requirements in the international maritime conventions lies with the shipowner/operator. Responsibility for ensuring such compliance remains with the flag state. We refer to the Paris MoU, the European port state control region's website[4], for further information.

A Port State Control in practice

The goal of a port state control is to assess the general condition of the vessel. The inspection is performed through an inspection officer coming on board when the vessels call a port. The

[1]http://www.skargardsredarna.se/om-oss/vad-ar-skargardsredarna accessed 2021-02-05
[2]https://www.stena.com/business/stena-line/ accessed 2021-02-05
[3]https://www.itopf.org/in-action/case-studies/case-study/amoco-cadiz-france-1978/ accessed 2021-02-05
[4]https://www.parismou.org/ accessed 2021-02-05

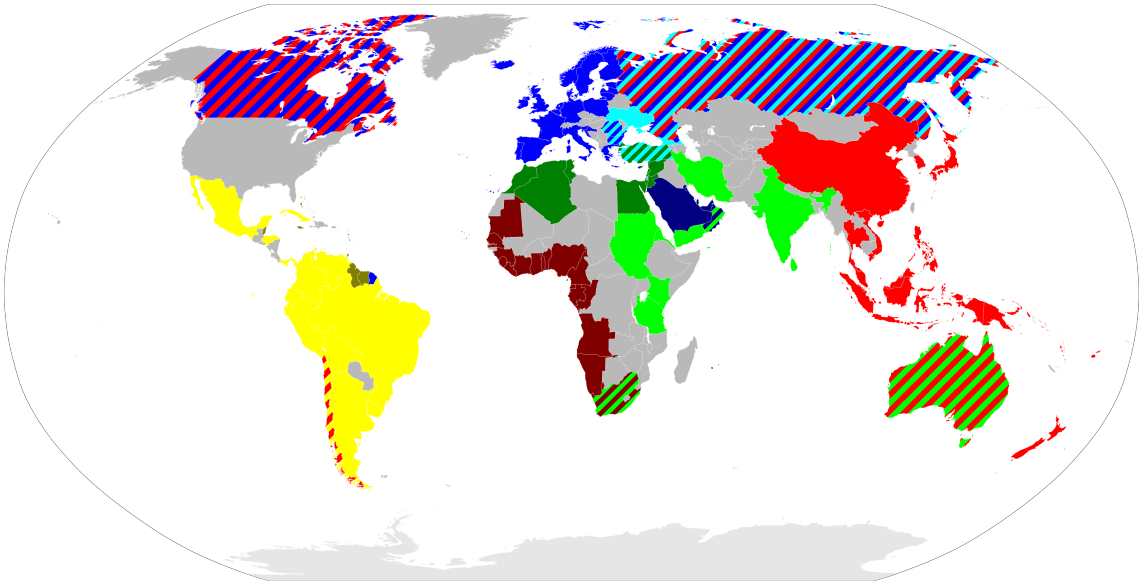

Figure 2.1: Signatories to the Paris MoU (blue), Tokyo MoU (red), Indian Ocean MoU (green), Mediterranean MoU (dark green), Acuerdo de Viña del Mar (yellow), Caribbean MoU (olive), Abuja MoU (dark red), Black Sea MoU (cyan) and Riyadh MoU (navy).

inspection officer, together with the master of the vessel, then together inspects the ship, working conditions, and documents. Due to the extensive target area of an inspection, only a subset is checked every time, with a special focus on specific areas depending on central directives and drives.

The result of the inspection is then saved using a standardized protocol. This protocol becomes available through an application on the Paris MoU web page. A deeper exploration and data relevant to the thesis is presented in Chapter 3.

2.1.3 Monitoring Reporting Validation (MRV)

The MRV system is a mandatory EU system that came into force on January 1st, 2018, to report CO_2 emissions from ships over 5000 GT calling ports in the European Economic Area (EEA). It was created due to the relatively slow progress from the International Maritime Organization (IMO), which led to the EU taking action. The reported data is centered around CO_2 and technical efficiency measures and is presented one year later on the THETIS-MRV page[5]. The MRV data is a dynamic dataset. However, only two years of data were available at the time of collection. Due to this, it can be considered a static dataset for our purposes. For more information regarding the MRV system, see the European Commission web page [6].

[5]https://mrv.emsa.europa.eu/#public/emission-report accessed 2021-02-06
[6]https://ec.europa.eu/clima/policies/transport/shipping_en accessed 2021-02-06

2.1.4 Clarksons Research

Clarksons Research is a commercial data and intelligence provider for the shipping, trade, off-shore, and energy industries. One of the services they provide is a searchable register of the world fleet containing detailed information about vessel characteristics such as size, engine types, number of propulsors, flag, ownership, and much more. In this thesis, the Clarksons data is referred to as static data because we had only the latest information available and thus could not create time-series data based on it.

2.2 Data Driven Analyses of Maritime Data

Several large-scale incidents, e.g., *Amaco Cadiz* incident mentioned earlier, with key issues being substandard vessels, led to the development of the current safety regime in shipping. This includes a complex, constantly amended global set of rules regarding safety, pollution, and vessel standards. One of the effects is the implementation of the Port State Control schemes on a regional level which has led to a standardization of data and processes that can be analyzed.

Over the years, the analyses started from statistical correlations, then binary logistic regression, and lately, Bayesian networks with often more extensive or specialized datasets. Below we will give an overview of the recent research in the field.

One notorious issue is the unreliability of incident reports and under-reporting in the maritime industry(Hassel et al., 2011; Psarros et al., 2010). Although this problem exists, many previous analyses have been done with successful results.

For an overview of previous empirical research on maritime accidents, see Bye and Aalberg (2018). In the following literature review, we focus on the data-driven approaches. One significant difference is that the number of variables used in these analyses is relatively limited compared to the scale of the dataset we will construct later.

Using Poisson modeling on data from 1999-2004 from the Swedish Maritime Administration, Cariou et al. (2006) is the first large-scale data-driven research we could find. It estimates how vessel characteristics influence reported port state control deficiencies. Based on this analysis, they show that repeated inspections reduce the number of deficiencies. The article also contains a good review of previous research done before the use of statistical approaches using large data sets.

Using a binary logistic regression model on 180,000 port state control inspections, Knapp and Franses (2007) measures the effect of inspections on the probability of casualty. They find that 43% of vessels can be identified as belonging to a group where inspections effectively decrease the probability of casualties. The strongest effect concerns the severe casualties, where they find a 5% decrease per inspection.

Using a smaller dataset of port state control inspections with a support-vector machine-based model, Xu et al. (2007) created a risk assessment system that classifies vessels as either high-risk or low-risk. A vessel is considered successfully classified if a high-risk assessment leads to a detention. Using this model, they show a relative improvement of nearly 50% from the baseline risk assessment accuracy achieved by the Paris MoU risk calculator used at that time. The article also contains some engineered features. With their assessment of accuracy as classified above, the accuracy for the Paris MoU risk calculator is 9%. With their support

vector machine-based model trained on data from both the Paris MoU region and Tokyo MoU region, they achieve an accuracy of 14%. Their accuracy is hard to gauge because they do not accurately specify how it is calculated.

Li et al. (2014b) created a quantitative ship safety index for each vessel in the dataset using binary logistic regression. They also use multivariate logistic regression to assess how various factors simultaneously affect the safety level. They conclude that the largest safety indicators are vessel type, size, classification society, specifically if not part of the International Association of Classification Societies (IACS), navigation zones, and registry type. The vessel types included in the study were in decreasing safety order *General Cargo*, *Passenger Container* and *Bulker* with *Tanker* being the safest.

The next step explored is using Bayesian network modeling. Li et al. (2014a) created a Bayesian network with priors created using binary logistic regression. The conclusion is that the ship's condition is the most significant single influencing factor on total loss occurrences, after that classification society and ship type.

Hänninen and Kujala (2014) created a combined dataset, although with different reporting periods, of Finnish port state control 2009-2011, Baltic Sea Accidents 2004-2010 from HELCOM, and Gulf of Finland VTS reported incidents 2004-2008. They conclude that the variables related to the type of ship, port state control inspection type, and structural-condition-related deficiencies are among the ones that provide the most information regarding accidents. Following this research, Hänninen et al. (2014) used a similar data set to model maritime safety management using Bayesian belief networks based on expert elicitation suggesting room for improvement. The strongest signal found regarding having an adequate overall safety management level was IT systems for safety management. They also find that if no deficiencies have been discovered in port state control, then the adequacy of the safety management is twice as probable compared to the baseline, even without knowledge of the inspection history.

Following the Bayesian track, Yang et al. (2018) used a tree-augmented Bayesian network. The model is validated using sensitivity analysis on port state control data coming from bulk carriers. They found that the most important risk factors are inspection group, the number of deficiencies, type of inspection, Vessel group, recognized organization, and Vessel age in the given order.

Using a tree-augmented naïve Bayes classifier, Wang et al. (2019) identified high-risk foreign vessels with a model trained on 250 records from Hong Kong. They did this by predicting the number of deficiencies on an incoming vessel and then validating it. They found that their method is better than the current risk selection scheme.

Taking a deeper dive into the port state control data and incident data to find more profound learnings, Bijwaard and Knapp (2009) researched port state control's effectiveness in prolonging ship lives. To do this, they constructed a dataset containing information regarding the timing of accidents, inspections, and ship particular changes on more than 50,000 vessels between 1978 and 2007. The conclusion is that there may be some over-inspection of vessels using this metric.

From another angle, Tsou (2019) used association rule mining, also known as market basket analysis, to explore detentions in Tokyo MoU. Finding much of the same as earlier studies, they also found more specific results. They found that fire drills and ISM have received more scrutiny since 2013, or at least their correlation regarding detentions increased. Also, geographical indicators were found on which deficiency areas correlate to detentions.

Following the general approaches, more specialized topics have also been investigated. Heij and Knapp (2015) investigated the effects of wind strength and wave height on ship incident risk. Using weather observations from buoys, ships, and other stations, they used binary regression to relate those to incidents. They found that more wind and waves have a statistically significant correlation to incidents in some areas for some ship types.

Using a multivariate logistic regression model, Bye and Aalberg (2018) investigated whether an accident is navigation-related or not and how ship particulars influence that in Norwegian waters. Finding that some vessel types, poor visibility, and flag of convenience increased the probability.

Concluding all this previous research, we see that statistical differences exist, often finding similar results in many studies. We see statistical differences based on size, flag, type, navigation areas, and classification societies. This validates using deep learning with more data as a possible approach for the problem space.

2.3 Deep learning and maritime data

During the preliminary investigation of the dataset, we found that combining two sub-fields in deep learning is required.

1. The first is the temporal aspect, ships age, and change, both in purpose, certifications, and ownership. This is coupled with inspections happening at varying intervals, creating a time series of a vessel's life.

2. The other aspect is that almost all available data is categorical, called qualitative or tabular.

Another factor that we will need to address is the variable length of the data; a vessel might have one inspection or, in other cases, 20. The problem can also be framed as anomaly detection. We will also investigate the available anomaly detection methods and their applicability.

2.4 Port state control risk index

Paris MoU has a ship risk profile calculator that is based on the concept of a simple scoring system coupled with some hard limits. For example, the flag state performance is a metric regarding how vessels flagged in that country have performed on previous inspections. There are three levels, white, gray, and black, where white is a good performing flag state. If a vessel is "black" or "gray", it can not be eligible for a low-risk profile, no matter the other data.

This is true for essentially all fields, where anything lower than the highest quality makes the eligibility for a low-risk profile disappear. Using the scores calculated, the weighing of the risk profile then assigns the vessel as one of three groups; *low*, *standard*, or *high risk*.

There also exists a *Company Performance Calculator*. It is based on the company inspection history from the last 36 months leading to a combined index, the *Company performance*, and to sub-indexes, the *Company Detention Index* and *Company Deficiency Index*.

The Company Performance Calculator works by taking the following fields:

- How many port state control inspections has the fleet undergone in the Paris MoU region?
- How many detentions have these inspections resulted in?
- How many Non-ISM deficiencies have been recorded during these inspections?
- How many ISM deficiencies have been recorded during these inspections?
- Has a refusal of access been issued to any ship of the fleet?

The calculator then calculates a ratio-based index based on the number of inspections and the entered data. The main *Company Performance* index gives four outcomes for the performance: *Very low*, *Low*, *Medium*, or *High*. The output is based on two sub-indexes, one regarding detentions and one regarding deficiencies. The output table is shown in Figure 2.2.

Company Performance

Detention Index	Deficiency Index	Company Performance
Above Average	Above Average	Very Low
Above Average	Average	Low
Above Average	Below Average	
Average	Above Average	
Below Average	Above Average	
Average	Average	Medium
Average	Below Average	
Below Average	Average	
Below Average	Below Average	High

Figure 2.2: Paris MoU company performance output table.

2.5 Applicable machine-learning methods

2.5.1 Random forest

In *Deep Learning for Coders with Fastai and PyTorch*, Howard and Gugger (2020) describes using Random Forests as a precursor to deep learning models. The reasoning behind this is the good performance on tabular data, ease of training, and lower sensitivity to hyperparameter choices. This allows for creating a good baseline, exploring data, and trying to pinpoint less impacting features leading to a simpler final model. An easy way to accomplish this is through feature importance ranking (FIR), built into scikit-learn. Out-of-domain outputs are issues with random forests: They cannot predict based on unseen data. For example,

if the output variable has a range of $5 \leq y \leq 20$ then the model cannot predict anything outside this range. A significant advantage of random forest models is that as long as the trees are kept reasonably general, they do not overfit. This was shown on *Bagging predictors* by Breiman (1996) with the following refinement when proposing random forests based on decision trees in Breiman (2001). They are also suitable for spotting any data leakage since it is straightforward to visualize and interpret the resulting trees.

2.5.2 Categorical/Tabular deep learning

Starting with *Survey on categorical data for neural networks* by Hancock and Khoshgoftaar (2020), we get an overview of available methods, broken down into categories and purposes. They identify three general categories of entity embedding techniques, although they are often mixed where the output from one is used in the next.

Determined, for example, "one hot encoding", where given the same dataset, the encoding will produce the same encoding every time.

Algorithmic, where the result may or may not be deterministic. Although it is generated before the training phase, they are more computationally intensive than determined.

Automatic, where the machine learning task automatically generates the entity embedding during the training phase.

Due to the already complex problem space, we start with a determined encoding and may move on to an automatic one later if it can be proven effective.

Determined

One hot encoding is a frequently used method. It assigns each category to a position in a vector and then has the position of the selected category be 1 while the rest is 0 (Hancock and Khoshgoftaar, 2020). The advantages are that it is straightforward to apply, takes minimal computational resources, and contains minimal artificial relations between the categories. A clear disadvantage is that it can create a truly enormous number of variables, which can lead to lower performance when training. This is somewhat mitigated using sparse matrix representations.

Label encoding assigns an integer value to every possible category, for example having three categories: $\{\text{"tanker"} = 0, \text{"passenger"} = 1, \text{"general cargo"} = 2\}$.The advantage is that it is extremely simple and compact. A disadvantage is that it introduces artificial ordering among the categorical variables and causes issues when using gradient descent due to each label contributing differently to the gradients (Hancock and Khoshgoftaar, 2020). For random forests, Wright and König (2019) find that on their three tested datasets it performs similarly to other methods.

Code counting , as defined by Hancock and Khoshgoftaar (2020), is based on defining some slice of the data and then counting the number of occurrences of a chosen value, parameter, alignment of something, or similar. The advantage is that it works for big data sets, while the disadvantage is that it inherently discards information. Therefore, code counting is not recommended to be used unless necessary.

Hashing is sometimes used with the clear advantage that the feature count stays the same, allowing us to work with big data sets more efficiently. The issue is if hash collisions occur. In *Deep Learning With Python*, Chollet (2018) describes the method. The same problem with varying gradients found in Label Encoding should also occur, although artificial ordering does not.

Algorithmic

Algorithmic methods are based on taking one of the determined encodings and then applying a fitting transformation to the output. This transformation allows the tuning of hyperparameters to fit the underlying data better (Hancock and Khoshgoftaar, 2020).

Hancock and Khoshgoftaar (2020) detail three techniques:

- Latent Dirichlet Allocation
- Generalized feature embedding learning
- Convolutional neural networks for categorical data

The main problem with algorithmic techniques is the computational power required and, hence, not suited for larger datasets. Due to this complexity, these methods will only be considered at a later step, given time.

Automatic

The main appeal of automatic techniques are that they are often more general and often allow for reuse. The issue is that they often require more computational resources when trained in the first place (Hancock and Khoshgoftaar, 2020). Therefore these will not be considered for this thesis due to the already high computational requirements.

2.5.3 Temporal deep learning

Fawaz et al. (2019), in *Deep learning for time series classification: a review*, present an overview of available methods for using deep learning on temporal data and test them on several available benchmarks. We will investigate their multivariate approaches regarding the maritime data's applicability. They describe two general approaches: *discriminative* and *generative*. The difference is that generative models usually exhibit an unsupervised training step preceding the learning phase. This type of neural network has also been referred to as *model-based* in the time-series deep learning community. The extra step creates extra complexity. A *discriminative* deep learning model is a classifier or regressor that directly learns the mapping between the time-series or its engineered features and outputs a probability distribution over the class of variables in the dataset. The informal consensus in the time series classification community is that discriminative approaches have higher accuracy than generative approaches (Fawaz et al., 2019). Due to this and the lower complexity of fewer steps, we limit ourselves to a discriminative model.

In their paper, Fawaz et al. (2019) tested proposed algorithms of multivariate data on the multivariate time series dataset from Baydogan (2020), concluding that a *Fully convolutional*

neural network (FCN) and *Residual network (Resnet)* perform the best. A central question remains: how applicable time series methods are to our data since the time steps have variable lengths and the data has exceptionally high dimensionality.

Fully convolutional neural network (FCN)

As proposed by Wang et al. (2017), FCNs are mainly convolutional networks with no local pooling layers. A pooling layer reduces the dimensionality of the data by combining the output of several neuron clusters into a single neuron in the next layer. The two common types used are *max*, which takes the max value in the cluster, and *average*, which takes the mean value. This has the effect that the length of a time series is kept unchanged throughout the convolutions. They also replace the traditional final FC layer with a Global Average Pooling layer. This drastically reduces the number of parameters in a neural network and enables the use of Class Activation Mapping (CAM). Using CAM is of great value in this thesis since it can give us a deeper understanding of why a specific classification was done.

Residual network (ResNet)

The Residual network proposed in Wang et al. (2017) is relatively deep with 9 layers followed by a Global Average Pooling (GAP) layer, which averages the time series across the time dimension. The `softmax` function is used for the output layer. An advantage is that the architecture has an invariant number of parameters across different datasets, except in the final layer. This makes it easier to do transfer learning which may increase the performance due to already recognizing common patterns in similar tabular data. This allows both better results and faster training due to less layers needing to be trained. The difference between a normal convolutional and a residual network is the shortcut residual connection between consecutive convolutional layers. This helps training by reducing the vanishing gradient effect (Fawaz et al., 2019).

2.5.4 Deep learning for anomaly detection

We know that most boats are safe, and we are searching for the anomalies, the unsafe boats. This means that the research problem of finding low-performing vessels can be phrased as an anomaly or outlier detection problem. In *Deep learning for anomaly detection: A survey*, Chalapathy and Chawla (2019), investigates the current state-of-the-art deep learning techniques or shortened as DAD, Deep Anomaly Detection. They give a good dataset size to change from machine learning to deep learning when it becomes gigabytes, which aligns with our case. Their review splits anomaly detection into two categories: sequential and non-sequential. We are interested in sequential anomaly detection.

In deep anomaly detection (DAD), the methods are split into three groups following the traditional deep learning approaches, *supervised DAD*, *semi-supervised DAD*, and *unsupervised DAD*. Supervised and semi-supervised methods are interesting since we have labeled data, which is usually more challenging to collect when detecting anomalies.

Supervised deep anomaly detection is, as it sounds, regular deep learning with labeled classes. A central issue is the imbalance of data: anomalies are by definition rare, and therefore, the data is imbalanced leading to sub-optimal performance (Chalapathy and Chawla, 2019).

Semi-supervised deep anomaly detection assumes that all training instances have only one class label and then finds outliers by testing examples not fitting to that class. The computational complexity is the same as supervised. The advantage is that only correct instances need to be labeled, which is much easier. Though not wholly applicable for us, it is an interesting avenue to test (Chalapathy and Chawla, 2019).

In their review, methods applicable to our problem space are Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs), where an LSTM is an RNN with specific properties. First, we will look at RNNs as a general concept and then explain where LSTMs diverge.

Recurrent Neural Networks (RNN) work by having a persistent state in the neurons. This is useful when classifying text by splitting it into, e.g., words, then feeding the words through, and the output will represent all the words fed through, regardless of the input length. The issue is that the beginning of the sentence ends up having less impact on the result than the end due to a problem similar to vanishing gradients, i.e., a recency bias on the input. In some applications, for example, text classification, the first words of a sentence often change the entire meaning. LSTM, *Long Short-Term Memory*, is a specialized method of implementing RNNs to solve this issue by having a more complex internal state. Depending on its impact, it allows vital information to be retained and passed to the next cell. This is what allows it also to have long-term memory.

2.5.5 Imbalanced data

The dataset we have is imbalanced and, in some respects, limited. There have been many methods developed to get around this problem. Fawaz et al. (2019) specify some methods to solve it, using a weighting of output classes or data augmentation. In anomaly detection, the data is inherently unbalanced. There has been some development using Generative Adversarial Networks (GANs) to generate samples for tabular data. Generally, it is a difficult question for tabular data since verifying that the generated samples are correct is tricky. Compare this to rotating an image of a dog. For us as humans, this intrinsically makes sense, and we can easily verify that the method is correct. Comparatively, what is the rotation of a sample of tabular data?

2.5.6 Deep learning conclusion

Working with tabular data, Howard and Gugger (2020). describe Random Forest as a good baseline to gauge information gain. As the research above shows, the following neural network-based methods seem applicable to our problem space.

- FCN
- Residual Neural Networks
- LSTM – Long Short Term Memory
- Recurrent neural networks

2.6 Validation

Validation of a novel model in a nascent field is a complex topic. We found only one comparable study by Xu et al. (2007)., which validated the data against the 2007 version of the port state control risk index. The model for the risk index is continuously updated and, therefore, can not be used as a fair comparison today since the available application uses the 2021 version of the risk index. We decided against trying to find the 2007 model since it would be of little relevance today. A more rigorous study with more time would develop a standardized test set to measure the performance and allow for future improvement similar to many other fields.

In *Machine learning algorithm validation with a limited sample size*, Vabalas et al. (2019) explored validation methods in technology-based data collection methods. Their study considered datasets containing small sample sizes with high dimensional data, which is a perfect analogy to this thesis' problem. The number of samples in the examples listed is sometimes small enough to force the real data to be used in the testing and validation. For those cases, they specify and test the available methods. Since we are not that limited in size, a regular train and validate split with defined random seeds should mitigate all these issues. The only remaining issue is keeping the validation dataset large enough to have enough statistical power to validate the model's accuracy.

As said earlier, random forest models are a good baseline that provides good results without normalization or parameter tuning. We will therefore compare all models developed against a random forest implementation.

2.6.1 Model accuracy

Accuracy is calculated differently depending on the target, implementation, and ratio of target classes. For the detention dataset, a binary classification, central terms used when calculating metrics are *accuracy*, *recall or sensitivity*, *precision* and *f-measure*. Central in all these measurements is the ability to take all the statistical outcomes listed below into account:

- True positive (tp)
- False positive (fp)
- True negative (tn)
- False negative (fn)

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{2.1}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{2.2}$$

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \tag{2.3}$$

$$\text{F-Measure} = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{2.4}$$

For the deficiency dataset, the problem can be modeled as a regression. Often used methods when calculating regression models' accuracy are mean absolute error (MAE) and root mean squared error (RMSE). MAE is closer to what a human would intuitively classify as the error since it measures how close the models' are to the truth instead of introducing weighting with roots and exponents. In the following equations, y_i is the prediction and $\hat{y}_i$ is the true value, and n is the number of samples. MAE and RMSE can also be used to give a percentage score by creating relative accuracy indices. Since this thesis focuses on interpretability, we will leave them out.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{2.5}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \tag{2.6}$$

Chapter 3

Data

This chapter describes the data sources investigated and the reasoning leading to the final selection. In tandem, the data collection is detailed together with the data cleaning performed. The chapter is laid out first to be an overview of the data sources, followed by the implementation of each dataset. Afterward, we explore the data and generate the datasets used to implement the models. The data pipeline used can be seen in Figure 3.1. The following description of the system has been simplified to describe the essential steps to reproduce the same result. We manually collected and saved the differences when we performed any removal due to data cleaning, allowing for manual sampling to verify the integrity of the process.

Generally, the data can be split into data and dynamic vessel data. Static data are vessel characteristics that generally do not change over time—for example, length, tonnage, and similar parameters. Due to limitations in the provided databases, we only have the most recent information. An example would be an inspection from four years ago coupled with the main engine being replaced this year. In our sample, this leads to the inspection from four years ago being counted as having the engine that was installed this year. Due to no possibility of correcting this and being equal for all vessels, this was left as is.

Dynamic data are vessel characteristics that change over time. Examples of dynamic data are owner, trade area, and machinery with shorter lifespans than the vessel itself. We used two sources for the dynamic data: MRV (Monitoring, Reporting, Validation) data and Paris MoU port state control inspections. In the following chapter, we will describe them in depth.

Due to the enormous number of fields collected, we only list the notable ones in the report. We also describe the general layout of the fields, which are often repetitive, with the difference being which certificate it refers to. This is mainly seen in the Paris MoU data.

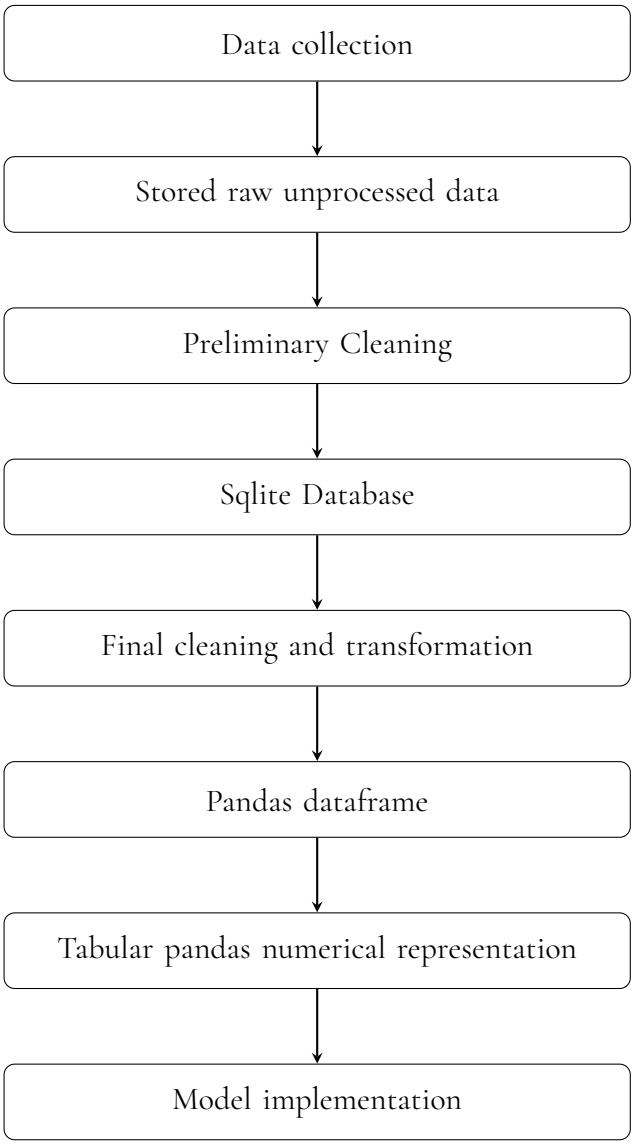

Figure 3.1: Flowchart for the data used.

3.1 Data sources

The selection criterion for data sources is whether it contains information regarding multiple vessels that we could use to assess their risk. To find applicable sources, we conducted a literature survey. A good starting point, although old now, is a 2012 thesis from Blekinge Tekniska Högskola (BTH) (Abghari and Kazemi, 2012), which has a comprehensive list of available sources at that time. We also noted down all data sources used when we explored the previous work in the field regarding data-driven analyses of the maritime industry. Finally, the last part came from discussions with domain experts at RISE. In the following section, the most notable data sources are detailed.

Clarksons Research Vessel Fleet Register. A searchable register of the world fleet containing detailed information about vessel characteristics. Clarksons Research

Vessel Fleet Register is an excellent quality data source with an enormous number of fields per vessel. Generally, the data quality becomes worse when looking up more specialized items. The quality is also lower for smaller vessels. Due to the quality and abundance of data, we decided to include the Clarksons Research Vessel Fleet Register.

IHS Markit. IHS Markit is a commercial data provider similar to Clarksons Research with similarly good data quality. Generally speaking, less data per vessel is available, but it is high-quality. Clarksons Research is more all-encompassing in comparison in that regard. Due to IHS Markit not being available for us when the project was started, initially, we decided not to use it. RISE switched data providers in the middle of the project. When issues related to exporting from Clarksons Research arose, we will describe in more detail in Section 3.2.1; we used IHS Markit to clear these issues, and it is therefore included.

Due to being forced to also export from IHS Markit, we decided to incorporate a few fields we found to contain better data than Clarksons Research. Those are listed in Table 3.5. In the following text, when the *Clarksons Research data* is referenced, the combination of Clarksons Research and IHS Markit is meant since they were immediately merged before any further analysis was done to solve the unique identifier issues.

Paris MoU. The Paris MoU port state control data is available through their web page. This is a very high-quality data source and is directly related to vessel quality, which is the organization's prime objective. It comprises about 18,000 inspections annually, going back to 2016. Based on this, we included this data in the final modeling.

Other MoU regions. Data is also available for the other MoU regions and from the US. The regions are shown in Figure 12. In studies, it has been found that the regions' quality of inspections varies greatly, where Tokyo and Paris have the best and about equal quality (Piniella et al., 2014). Another issue is that some regions publicize their complete inspection protocols, others only detentions, while in one case, the entire website had even stopped working at the time of consideration.

Based on this, the Tokyo data was considered the most closely related. Nonetheless, we decided to leave that out as future work and improvement.

MRV - Monitoring Reporting Validation. The MRV system is a mandatory EU system that came into force on January 1, 2018, to report CO_2 emissions from ships over 5000 GT calling ports in the European Economic Area (EEA). The reported data is centered around CO_2 and technical efficiency measures and is presented one year later. This is a dynamic dataset, although since only two years of data are available, it is essentially static for our purposes. Due to the reasonably good quality data and possible interesting connection between safety and environmental questions, we decided to include this dataset.

EMCIP. EMCIP is a European database containing summaries and links to the complete reports of the most severe accidents and incidents. A scraper for the EMCIP database was developed. Due to the limited number of incidents, different time frames compared to other databases, and hard-to-use data due to less standardization, we decided not to use the dataset after it had been collected. Another issue is that many incidents that it describes are severe

enough to lead to the complete loss of the vessel, which means they do not show up in the Clarksons Research dataset, which only lists active or laid-up vessels.

Clarksons Research Incidents. The Clarksons incident database contains smaller and larger incidents as reported by shipping companies. The dataset size is usable: 9878 incidents concerning 7665 vessels beginning in 2016. The main reason to not incorporate this dataset in the model is the increased complexity it would result in. The reason for the increased complexity is because the time variable does not align with with the PSC reports which will be used as the discrete time steps. The Clarksons Research incident database would therefore require a larger effort to feature engineer it into something useful.

There is also the issue of dataset imbalance since not every vessel has incidents, and some has multiple ones. I we want to find the intrinsic parameters of operation leading to safer vessels, then a reported incident in the previous time period could likely be the by far strongest signal drowning out the other more important ones.

A method we thought of to handle this would be to create statistical analyses out of for example vessel sizes, types and so on which previous research has found to correlate with incidents then add as engineered features to the samples. Another method would be to construct a model predicting if an incident would happen before the next inspection. This could then be used to construct a combined model together with the inspections for a more comprehensive analysis.

Due to imbalanced data, complexity, feature engineering requirements and time constraints, we together with my advisers at RISE decided to not implement this and leave for future work in the area.

IMO GISIS Incidents. This is essentially EMCIP but on a global scale, containing more samples but less data per sample. If used, a solution for duplicated items would have to be developed due to many of the same incidents existing in both the EMCIP database and Clarksons Research incidents database.

IMO GISIS Piracy. Database containing piracy attacks globally. This is also a limited dataset. We did not see a clear connection between operating the ship in a certain way coinciding with risk and piracy attacks due to the geographical hot spots and the opportunistic environment piracy attacks occur. These would also need to be correlated by date to inspections leading to a more complex dataset. Based on this, we decided not to include the piracy database.

ILO Breaches. This database contains shipping companies breaching the International Labour Organization (ILO) agreements. These companies do not follow the most basic requirements for workers in a global low-cost industry. We found this database later in the project. Due to this thesis using vessel samples and the ILO database naming companies, a correlation essentially by hand would have to be done to include it. Therefore, we decided not to use it.

3.2 Data collection

The data was collected using the same API endpoints as the available public interfaces or exportation tools. The following sections will detail the methods and issues faced when collecting the data.

A central theme of the project was dealing with extremely messy data. This meant cleaning happened at essentially every transformation. For every cleaning, as a first step, we manually looked at the data to see if it was possible to apply some automatic transformations to fix the broken data. We either scraped the entire sample or left it as a missing value depending on how severe the issue was. Due to the large size of the dataset, this data cleaning should not impact the results achieved in a significant manner. Through the description of the processing and collection from each source, we will describe the decisions made regarding this and the feature engineering done for the final dataset.

3.2.1 Clarksons Research

Collection

Data from Clarksons Research was collected using their exportation tool and selecting the desired fields. An issue was that the exportation of IMO Numbers, the chosen unique identifier, was limited after a certain number of exports. To get around this issue, we used the MMSI numbers instead that were unrestricted. These were then linked to the original samples using the IHS Markit database, which allows for unlimited exportation of both. Before settling on the IHS Markit database, we tried using online available AIS (Automatic Identification System) data by scraping a public provider. This was deemed unworkable due to the extremely low quality of the data collected. AIS data is manually entered into the AIS devices by the seafarers. In another attempt, we created a scraper using `Selenium`, a library that allows makes it possible to control a web browser through code, and `Beautifulsoup`, which parses HTML data into a structured form, for the IMO GISIS page, which is the provider of IMO numbers. This was hindered by a limit of 100 searches per day.

An issue that can arise with this method is that IMO numbers are static for the vessel's entire life, while MMSI numbers are unique but do change when the vessel changes flag. Due to about a week between the collection from Clarksons Research to the cross-referencing from IHS Markit, some vessels may have changed MMSI. This is extremely limited, though, and should have no discernible impact on the final dataset given the long cycle times of the maritime industry, with shipping companies owning vessels for periods spanning years or decades usually.

The Clarksons Research data is generally of excellent quality, although going deeper into the dataset with more specialized features detailing the ships, the quality decreases significantly. This is represented by the percentage of missing numbers increasing and sometimes irrational numbers being given. Another limitation was a maximum of 40 columns exported per time. Due to this, we completed several rounds of exporting data.

Table 3.1 shows a sample of the most interesting fields from the 184 collected from Clarksons Research.

Clarksons Research Fields
Built Year
GT
LOA (m)
Beam (m)
Dwt
Builder Country/Region
Country/Region (Builder Group)
Group Owner Country/Region
Group Owner Full Company Name
Group Owner Nationality/Region

Table 3.1: Most interesting Clarksons Research fields .

Cleaning and dataset generation

We encountered several issues with the Clarksons Research data. The first issue we encountered was malformed CSV files exported by the platform. The exported data did not completely conform to RFC4180, the CSV standard, regarding escaping apostrophes. Therefore we needed to clean it. This was dealt with by creating a regular expression solving the issue and applying it to the raw files before anything could be further analyzed.

Due to the mentioned issue with the unique identifiers and columnar export limits, we developed a script to join all this entangled data spread over many files and fields to a coherent database. This allowed us to specify which identifier to use and then make joins adding more columns as they came while not making copies if the same data had already appeared. This is due to a human error leading to multiple exports of the same data. To validate that no errors had been introduced, we looked up a sample of vessels, comparing the database directly with the source data, and we did not find a single deviance. Thus, we can conclude that we successfully joined all data into a complete dataset.

Due to having few columns, we cleaned on a column-by-column basis, looking at the data provided. For example, flags in the database are not consistent between fields. Sometimes they are true or false. Sometimes a missing flag represents not having that item; sometimes, they are misspelled strings. For example, both “Y” and “y” signifies true, and so on. In Pandas, we used the function `.value_counts()` to ensure all cases were treated similarly between the samples.

To exemplify the issue, take the field `Number of Ramps`, in which missing means no ramps. In this case, a categorical value representing none does not work since it is an ordered discrete variable. Therefore we decided to fill all missing values with 0 assuming the missing field means the vessel does not have any ramps. Also, in the same field, there are human data entry problems, leading to some vessels having more than 10,000 ramps. Issues like this were handled by setting these values to missing, making them flagged as missing when moving on to the deep learning approach.

Another encountered issue stemming from the IHS Markit data was one column using a comma as a thousand’s separator. This led to the column being treated as categorical instead of continuous, thus needing a fix.

3.2.2 Paris MoU

Collection

Due to the relatively large size of the Paris MoU dataset, over 100,000 inspections on 29,510 vessels beginning in 2016, we found that the previously developed naïve slow scrapers would take too long. Due to this, we created a rate-limited asynchronous scraper using the Python packages `asyncio` and `aiohttp` to parallelize the task. We performed the scraping in two steps due to the layout of the API. First, we collected all the unique inspection IDs. These were then used when requesting detailed inspection reports.

We collected the into a large JSON file, the format given by the application, which was then processed. The large file size made loading it all in memory impossible on a regular laptop. Therefore lazy loading using the package `ijson` was done to circumvent this issue.

The data includes performance figures regarding the flag. Due to this being a derivative from the same data and would influence the model strongly, we decided not to use that field and instead see if the same trend could be seen directly from the data. The fields available per inspection are described in Tables 3.2 and 3.3.

The data is described using a standardized set of codes for the deficiencies. The first part of the number describes which group a deficiency is in, followed by code more accurately describing the nature of the defect. Listed below are some samples. For a complete description, see the list of deficiency codes on the Paris MoU web page [1].

- `01 - Certificates & Documentation`
  - `01117: International Oil Pollution Prevention (IOPP)`
  - `01201: Certificates for master and officers`
- `07 - Fire safety`
  - `07102: Inert gas system`
  - `07114: Means of control (opening, pumps) Machinery spaces`

Cleaning and dataset generation

Due to each inspection being different, some being thorough and detailed while others being quick and superficial, we decided to make the cleaning inclusive. Therefore `None` values were inserted for empty fields.

We encountered trouble with human data entry, mainly through dates being unreasonable. We took steps to fix the easily spotted errors, for example converting 217 to 2017 in a year field and similar, which is reasonable given that, for example, the certificate is valid for five years. We decided to discard the samples containing ambiguous dates.

For the certificate and deficiency data, we introduced a categorical value to represent the vessel not having it, since otherwise those values will be dealt with as missing, which is incorrect.

[1] https://www.parismou.org/list-paris-mou-deficiency-codes, accessed 2021-02-10

Ship details
General data about the vessel
IMO Number
Name
Flag
Type
Gross Tonnage
Keel Laying date
Age
ISM Company
IMO Number
Name
Address
City
Country
Class Certificates
List of class certificates issued.
Class certificate
Issue date
Expiry date
Statutory Certificates
Internationally recognized certificates inspected.
Statutory Certificate
Issuing Authority
Issue Date
Expiry Date
Surveying Authority
Last Survey Date
Last Survey Place

Table 3.2: General port state control inspection protocol part 1.

Another issue we encountered was incorrect IMO numbers. We removed those automatically by choosing to base the merge on the Clarksons Research data.

Lastly, the size of this dataset makes it extremely troublesome to work with in memory. The project had to be migrated to the RISE compute server with 192 GB of RAM, giving ample margins. Even this server crashed due to running out of memory, leading to further complications and time spent working around these issues when constructing the dataset.

3.2.3 MRV

The MRV dataset is easily collected in XLSX format from the THEIS MRV page. Table 3.4 lists the most interesting fields available.

The data is of good quality with few human errors. For some fields, missing equals zero, which we filled in. On others, zero meant that the data was unavailable, so we converted

Inspection details
General information regarding the inspection
Type of Inspection
Place of Inspection
Date of First Visit
Date of Final Visit
Number of Deficiencies
Number of Deficiencies Ground for Detention
Inspected Areas
List of areas which was inspected
E.g. "Cargo area", "Navigation bridge", etc.
Operational Control
List of operational drills performed.
E.g. "Emergency steering drill", "Fire drill", etc.
Deficiencies
List of deficiencies found during the inspection.
Area
Defective Item
Nature of Defect
Ground for Detention
RO Related
Accidental damage
ISM Related
Detention details
If detained, information regarding it.
Reason
Allowed to proceed to agreed repair yard
Date of detention
Date of Release

Table 3.3: General port state control inspection protocol.

MRV Fields
Technical efficiency measure (EEDI or EIV)
Technical efficiency number
Verifier data
Method of data collection
Annual time spent at sea
Annual average Fuel consumption per transport work

Table 3.4: Most interesting MRV fields available.

them to missing. We added a year suffix to the columns to allow for several years to be used in the sample.

Cleaning and dataset generation

Generally, there was no trouble with the MRV data. Some binary flags were represented with "Yes", which we easily fixed. There were some human-readable string representations for the technical efficiency in which we developed regular expressions to parse the correct data from the supplied string. To validate the correctness of this, we manually verified a sample of the results.

3.2.4 IHS Markit

The primary use of IHS Markit was to resolve the issue of the missing unique values from Clarksons Research due to them not having exportation restrictions on unique identifiers. Similarly to the Clarksons Research data IHS Markit also restricts the number of fields that can be exported from the tools simultaneously. Since only MMSI and IMO were necessary to resolve the issue, we decided not to skip available data related to the safe operation of vessels. The extra fields exported are listed in Table Table 3.5. The extra fields from IHS Markit, shown in Table Table 3.5, were included in the Clarksons Research data before any processing was done. They can therefore be considered as one dataset.

IHS Markit Fields
Vapour Recovery
Thruster Largest Type
Thruster Number
Thrusters Total kW
SMC Auditor
SMC Issuer
Crew

Table 3.5: Extra IHS Markit fields exported.

3.3 Data exploration

In this section, we will describe the exploration of the dataset used to gain a better understanding leading to the implementation of the models. Generally, due to the large number of features, it is hard to capture all-encompassing pictures with graphs. However, we try to show the similarities to data represented by the maritime articles described in Chapter 2.

In Figures 3.2, 3.3, and 3.4, some abbreviations for the vessel types were used. For a description of them see Table 3.6.

During the exploration, as anticipated, a highly imbalanced dataset was found with large cardinality. For detentions, only 3.4% of the inspections lead to a detention. The tail is summed into one bar at the end to keep the charts readable and present the most important data..

The distribution of the number of deficiencies in inspections leading to a detention is visualized in Figure 3.7. This is contrasted with the distribution of deficiencies not leading to a detention in Figure 3.6.

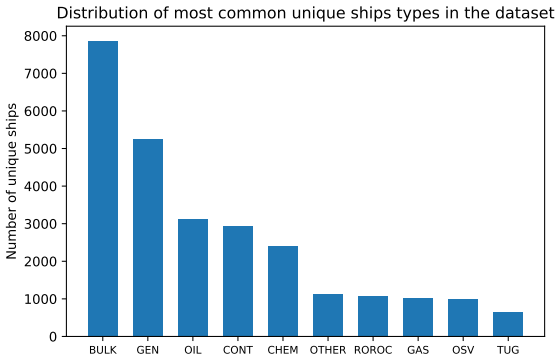

Figure 3.2: Distribution of ship types by unique vessels. Refer to table 3.6 for a description of the abbreviations.

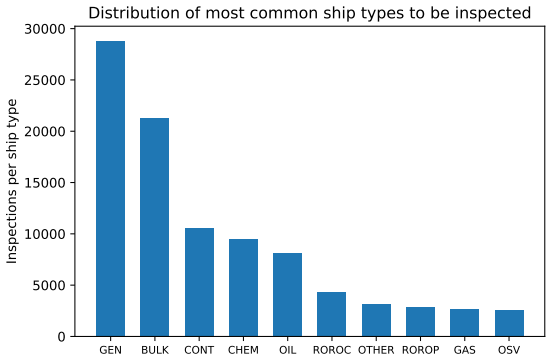

Figure 3.3: Distribution of ship types by inspections. Refer to table 3.6 for a description of the abbreviations.

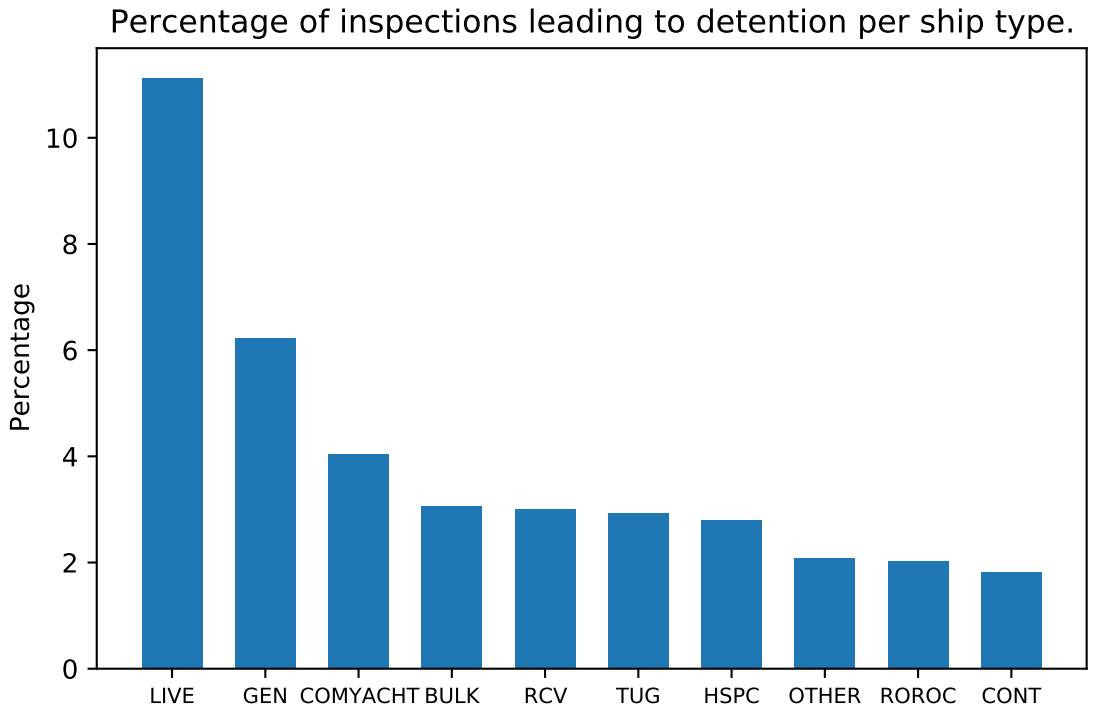

Figure 3.4: Percentage of inspections leading to detentions for ship types with more than 400 inspections. Refer to table 3.6 for a description of the abbreviations.

Seeing this difference, we explore the inspection preceding the detention. The result is shown in Figure 3.8. As seen, we have a shift to more deficiencies, the mean increases by 3.37 deficiencies.

Another imbalance is the number of inspections per vessel. This is explored in Figure 3.5, showing a large trend towards the lower side.

To further explore the collected data and better understand the problems facing the implementation, we generated a derivative dataset calculating the values in Table 3.7. We used

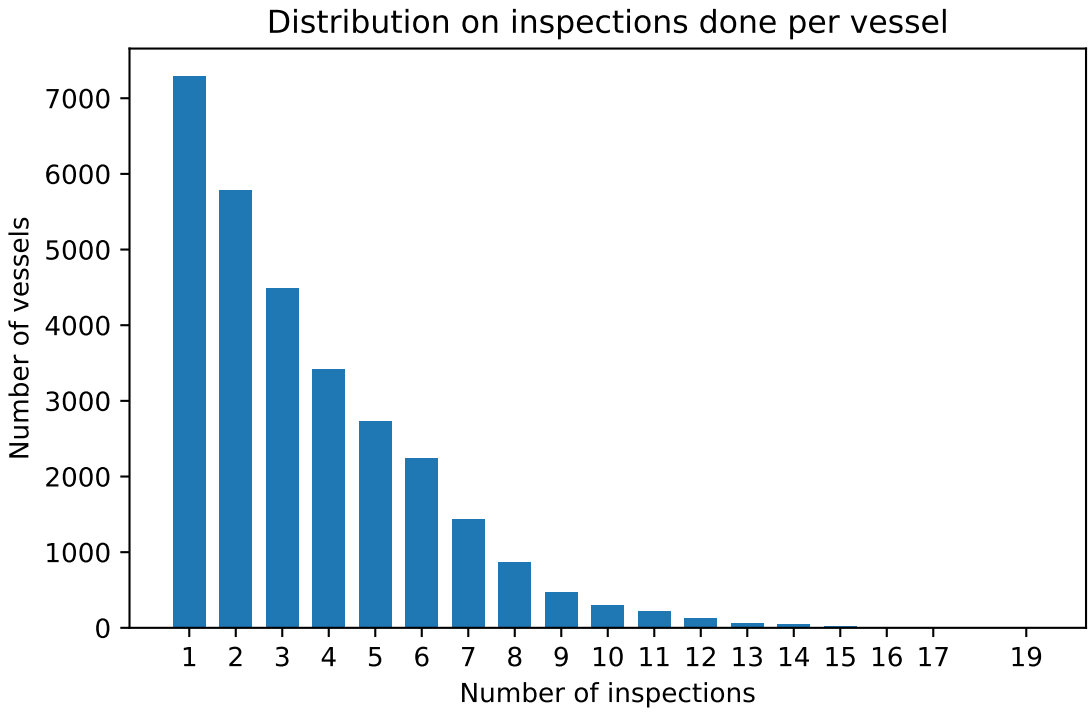

Figure 3.5: The distribution of the number of inspections per unique ship in the dataset.

Abbreviation	Description
BULK	Bulk Carrier
CHEM	Chemical tanker
COMYACHT	Commercial Yacht
CONT	Container vessel
GAS	Gas tanker
GEN	General cargo
HSPC	High Speed Passenger Vessel
LIVE	Livestock carrier
OIL	Oil Tanker
OSV	Off-shore support vessel
Other	Other
RCV	Refrigerated Cargo Ship
ROROC	Ro-Ro Cargo
ROROP	Ro-Ro Passenger
TUG	Tug

Table 3.6: Abbreviations used in figure 3.2, 3.3 and 3.5.

this data to guide the engineered model, focusing on limiting cardinality and missing values.

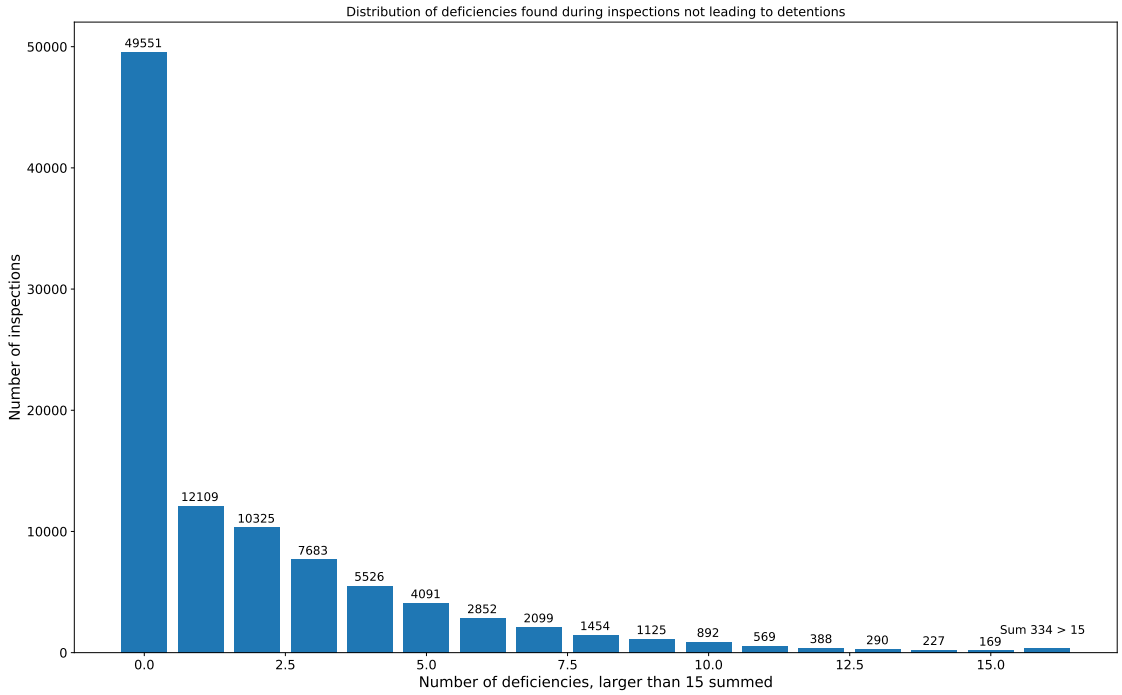

Figure 3.6: The distribution of the number of deficiencies found per inspection not leading to a detention. Mean is 1.89 with a standard deviation of 2.93.

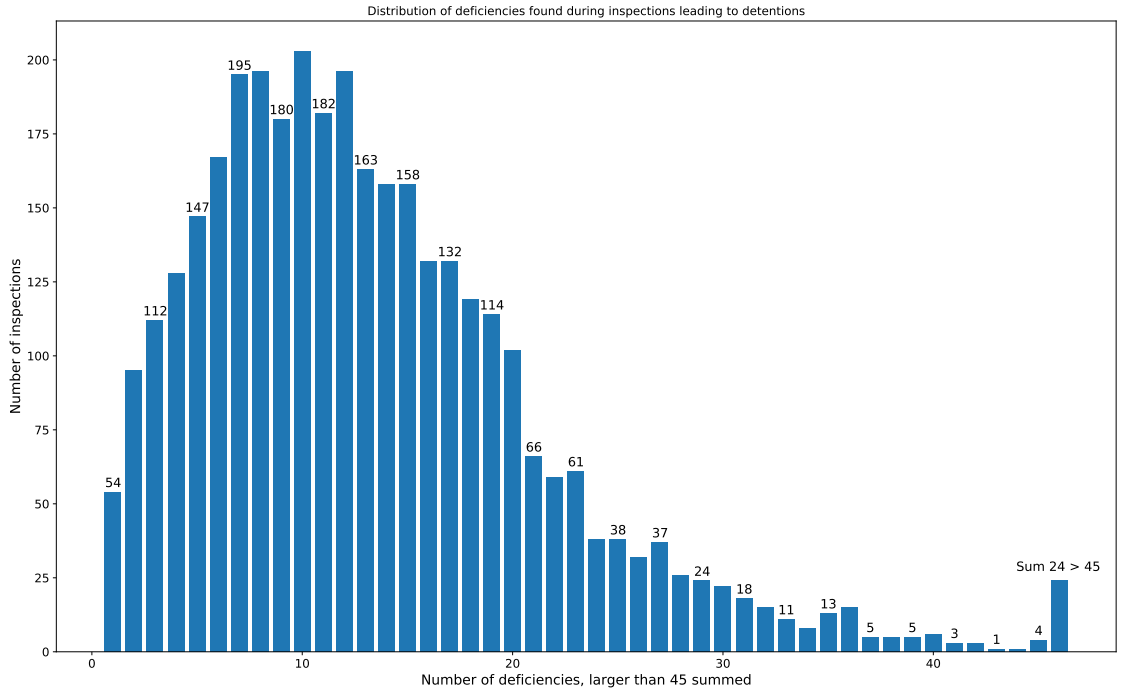

Figure 3.7: The distribution of the number of deficiencies found per inspection leading to a detention. The mean number of deficiencies is 13.40 with a standard deviation of 8.68.

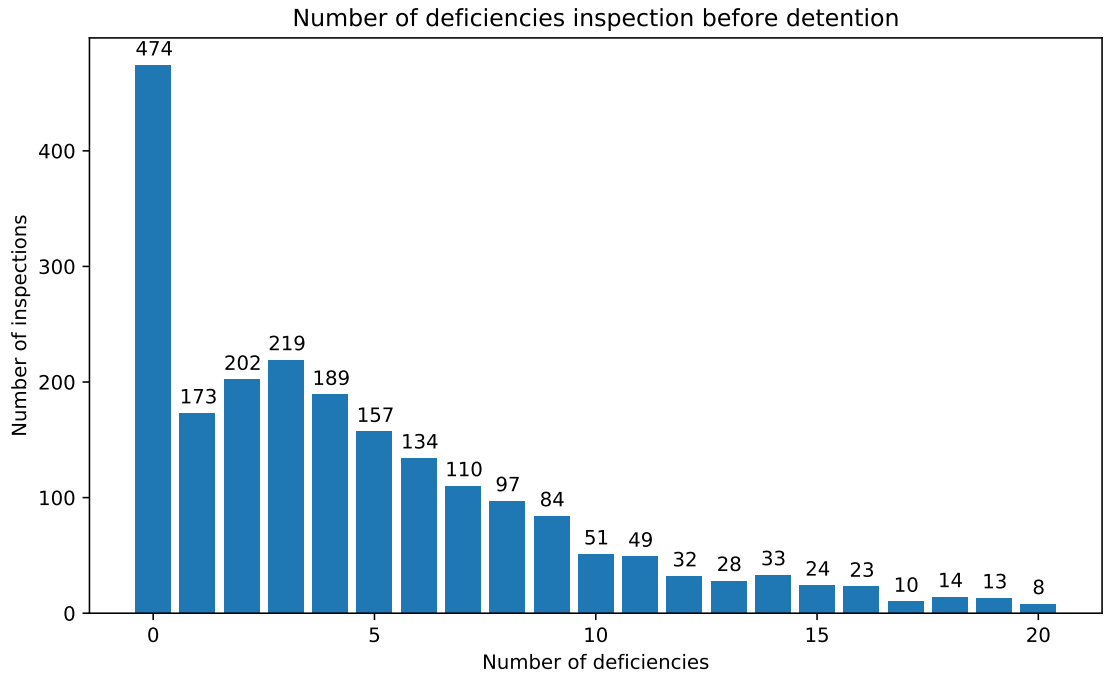

Figure 3.8: The distribution of the number of deficiencies found in the inspection preceding a detention. Mean is 5.06 with a standard deviation of 6.08.

Variables
Cardinality
Total number of Entries
Number missing
Percentage missing
Min
Max
Mean
STD
1st most common value and corresponding number of entries
2nd most common value and corresponding number of entries
3rd most common value and corresponding number of entries
4th most common value and corresponding number of entries
5th most common value and corresponding number of entries

Table 3.7: Datapoints per variable generated to explore the dataset.

3.4 Data imbalance

Based on the complex question of generating new data as described in Section 2.5.5, the proposed method is to make do with the collected data. One identified solution for the detention dataset, which by far is the most imbalanced, is to use a ratio, thus limiting the size of the dataset.

To allow for a good amount of data, we decided to use an 80:20 ratio. This allows the neural network to work with more information while having a reasonable split of the output classes.

To validate the aforementioned 80:20 split, a dataset using a 50:50 split is also proposed.

For the validation set, we propose to use a 50:50 split for all detention-related datasets to keep the methods comparable. This lowers the usable data but is a worthwhile trade-off not to introduce more uncertainty and create equal comparisons.

3.5 Combined dataset construction

This section will first describe the general aspects of constructing the combined dataset regarding the categorical encodings, time aspect, and targets. Following this, the implementation of the simple and engineered dataset is described.

We constructed the final dataset by joining the three earlier mentioned sources based on IMO number through the following steps:

1. Left join the Clarksons Research and MRV data, using the Clarksons Research as the base. This is called the static data.

2. Inner join the static data with the Paris MoU data.

The dataset size led to issues reaching the barriers of time and complexity that Pandas could handle. Following the proposed simple and advanced implementation method, we generated two datasets with different targets allowed.

3.5.1 Categorical encoding

Following the proposed methods for categorical encoding in Section 2.5.2, we tried one hot encoding first. For the simple dataset, this led to over 200,000 features per sample and was deemed unusable with the current infrastructure. Due to already being implemented in the tools used and having good performance for random forest models, we used label encoding instead. Finally, we resorted to code counting for the engineered dataset to keep it workable.

3.5.2 Time aspect

With enough complexity in the project, we decided to stay clear of variable length inputs. We can vary the time lag, although this limits the dataset. A time lag of one requires at least two inspections per vessel, allowing one to be the target; a time lag of two requires three, and so on. This would allow for the cleaned dataset sizes in Table 3.8 from the initial data of 103,160 samples. The mean number of inspections per vessel is 3.50, median 3, STD 2.46, min 1, and max 19.

For real-world applications, the time lag needs to be as short as possible to reduce the time to value for the model. For each consecutive inspection a vessel has to go through before being considered, the value of the model decreases. We choose a time lag of 3 to balance the need for data with time to value.

Time lag	Samples available
1	62828
2	44253
3	34966
4	20122
5	12728

Table 3.8: Number of possible samples depending on time-lag chosen.

This led to issues with the complete MoU dataset being too large for Pandas. Database operations did not finish within a reasonable time. Due to time constraints, the project could not be transferred to parallelized or distributed solutions. This is what led to the engineered dataset using code counting.

A lot of the data is dates, for example, when certificates are issued, their expiry dates, and the build date of the vessel. This is of interest to the model. An example would be having a certificate with a five-year running length. This would mean the part was up to the required specification when the certificate was issued. The question is if the part has deteriorated since and if there are more profound correlations within the dataset.

To capture these correlations, dates can not just be a label-encoded categorical variable. There need to be relations between them. To handle this, they are converted to mathematical representations. The variables are listed in Table 3.9.

Date transformation	Sample (2019-12-04)
Year	2019.0
Month	12.0
Week	49.0
Day	4.0
Dayofweek	2.0
Dayofyear	338.0
Is_month_end	False
Is_month_start	False
Is_quarter_end	False
Is_quarter_start	False
Is_year_end	False
Is_year_start	False
Elapsed	1575417600

Table 3.9: Encoding of dates to allow for differences to be detected.

3.5.3 Bucket target

Instead of regression analysis, we can sort the deficiency numbers into buckets. This changes the problem from regression to classification. From the means and standard deviations given for the inspections leading to detentions and not detentions in Figures 3.7 and 3.6, we propose

three buckets as given in Table 3.10. These were implemented and tried, both on the simple and engineered dataset.

Range	Interpretation	Percentage of inspections
0	Stellar condition	48.0%
1 – 5	Passable condition	39.0%
6 – ∞	Worried condition	12.9%

Table 3.10: Proposed distribution of buckets for vessel classification.

3.5.4 Missing values

These datasets were explored to get as much information as possible and validate earlier proposed methods. To handle missing values, we used the default implementation. This sets the missing value as the average of all existing values in that variable and adds a binary flag set to true, representing a missing value. For the engineered dataset, cut-off points were introduced. The process leading to that is described in Chapter 3.6.2.

3.5.5 Cleaning and validity concerns

There are always validity concerns when cleaning data. On the other side, no dataset is perfect, so cleaning is a required task when working with data. To keep the validity of the dataset, we performed the cleaning manually, trying to solve easily spotted errors or simply labeling the value as missing, thus still allowing the sample to use its other characteristics.

Starting with 103,160 inspections on 29,510 vessels, we end up with 62,828 samples. This, coupled with the fact that 7,292 vessels only have one inspection, means that we have lost 3,530 samples or 3.4% due to the cleaning performed. This number is not insignificant but should not impact the validity of the results.

3.6 Implemented datasets

With the final dataset available and transformed, two datasets were implemented: A simple dataset and an engineered dataset incorporating time lag. Following, we describe the implementation details for both datasets. The detention models have the most limited number of samples available. To alleviate this issue, a method to vary the ratio of the training set was implemented. This was used both for the simple and engineered dataset. This allows a 20:80 split between detentions and not detentions, which should make the model generalize better. For the validation set, we kept a 50:50 split allowing everything to be measured and compared equally.

3.6.1 Simple dataset

The first model we implemented was simple, without any feature engineering or selection, with the explicit goal of validating the entire pipeline from collected data to predictions. To

keep it simple, we used a time lag of one. For a real-world use case, this is the one with the most applicability since it allows for the model to be deployed relying on only one initial inspection when the vessel first enters the Paris MoU area.

This dataset contains all the deficiency-related data, as given in Table 3.3. For certificates, all data is contained regarding when it was issued, its issuer, and its expiry date. This is described in Table 3.2. One hot encoding was tried but deemed unfeasible on the dataset due to the high cardinality of many columns. The number of columns was in the magnitude of 100,000s, and the pre-processing steps had to be canceled after running a day without visible progress. Due to this, we decided to use label encoding instead since that means we do not increase our column count.

Using one inspection and then predicting the next, a time lag of 1, we reduce our entire dataset of 103,160 samples to 62,828 samples. Each raw sample contains 7051 columns. After the missing values have been dealt with as described earlier, each sample contains 7913 columns.

3.6.2 Engineered dataset

When first trying to implement time lag using the simple method, only stacking the columns after each other and sorting the values, a dataset containing over 23,000 columns was created. This was unworkable with the current infrastructure. Given time constraints, trying a more distributed approach was impossible. Therefore, a simplified dataset with as much grouping discarding as possible was created. To accomplish this, we discarded columns containing a large fraction of missing values. For the Clarksons Research dataset, the breakpoint was chosen at 90% and for the MRV data at 30%.

We used the code counting method described in Chapter 3 to group the deficiencies following the same nomenclature used in the Paris MoU system. With this method, we reduced the size of the simple dataset from 7913 columns to 2195 columns, giving a final dataset of 34,966 samples and 5235 columns with a time lag of 3. Notice the nonlinear scaling of columns because only inspection data is added per inspection, while the static vessel data is not repeated.

To further reduce the dimensionality, we tried using a principal component analysis (PCA) to project the dataset into a smaller dimensional space. A central issue with PCA is the loss of the previously defined variables, which lowers interpretability.

Due to time limitations, we only feature-engineered the features listed in Tables 3.11 and 3.12. The feature engineering we did was adding columns containing scaled variables to the engineered dataset. An example is the variable `Auxiliary Derived Total Generated Combined kW` which is essentially entirely dependent on vessel size. A larger vessel has more generators to power more, e.g., refrigerated containers. Left alone, it is simply another indicator of size. Dividing this number by the Dead Weight Tonnage (DWT), we get a more comparable ratio between vessel sizes. This should allow a model more readily to distinguish if a vessel is over or underpowered in this category.

Fields
Group deficiency count
Group ISM Deficiency count

Table 3.11: Simplified deficiency data

Variable	Scaled by
Auxiliary Derived Total Electrical Generated (kW)	DWT
Auxiliary Derived Total Generated Combined kW	DWT
Auxiliary Derived Total Mechanical Generated (kW)	DWT
Engine Derived Total Electrical Generated (kW)	DWT
Engine Derived Total Main Engine Mechanical kW	DWT
Engine Derived Total Mechanical Generated (kW)	DWT
Engine Derived Total Mechanical Propulsion (kW)	DWT
Hatches Total No	LOA (m)
Holds Total No	LOA (m)
Main Engine 1 (mkW)	DWT
Main Engine 1 Bore	DWT
Number of Ramps	LOA (m)
Shaft Generator 1 (ekW)	DWT
Thrusters Total kW	DWT

Table 3.12: List of scaled variables

Chapter 4
Model Implementation and Results

In this chapter, we will explore the implemented models, their results, and the issues encountered. For a succinct presentation of the results only, see Chapter 4.4. The random forest models pertain to **RQ1**, the baseline models used to verify our results, and the deep learning models pertain to **RQ2**.

For all detention models, no matter the split of the training data, the test set was always a 50:50 split between the target categories to allow for equal testing.

RISE provided the infrastructure to run the models, a server containing 192 GB ram, 8C/16T CPU, and three 1080 TI GPUs. For this thesis, most benefits came from the large amount of ram, allowing us to keep working in memory and the compute power given by the GPUs.

4.1 Simple dataset

We used the simple dataset to validate the pipeline using a broad selection with no feature engineering or time complexity. Table 4.1 contains the results.

4.1.1 Detentions

Random Forest (RQ1)

We implemented the simple random forest model using `sklearn` with a 50:50 split on detentions and non-detentions. Model parameters as given in the code below.

```
m = RandomForestClassifier(n_jobs=-1,
                           n_estimators = 1600,
                           max_samples = 200,
                           max_features = 0.5,
```

```
                            min_samples_leaf = 0.0025,
                            oob_score = True)
```

For the 80:20 split, it took a lot of work to get the Random Forest model to correctly detect the detentions, which essentially become outliers using this dataset. The parameters are given below, and we could not get past the baseline by tuning them.

```
m = RandomForestClassifier(n_jobs=-1,
                            n_estimators = 800,
                            max_samples = 800,
                            max_features = 0.5,
                            min_samples_leaf = 0.0025,
                            oob_score = True)
```

Model	Accuracy	Precision	Recall	F - Measure
RF 50:50	0.722	0.756	0.659	0.703
RF 80:20	0.522	0.892	0.051	0.096

Table 4.1: Scoring of random forest detention models. The split 80:20 split model has a hard time capturing the data.

Deep learning (RQ2)

We implemented the deep learning model using `fast.ai`, a front-end for Pytorch. As with the random forest model, we used a 50:50 and 80:20 split. Generally, when implementing the model. Initially, we had difficulty getting the models to converge, no matter the hyper-parameters used. This was likely caused by the large number of features compared to samples.

That was fixed by setting `y_range = [0, 1]`. The next issue we encountered was that the model quickly became overfitted, leading to poor generalization. The method to prevent this was a combination of a high weight decay and a low learning rate. This made the training take longer because more iterations were needed to capture the information in the samples, but it did provide better results.

We trained the 50:50 model on 4334 samples, with 30% used for validation. The code used is listed below. Generally, it took much work to find stable parameters for the deep learning model, which would match up to the random forest model. For the 50:50 model, it took much work to come above 0.67, not quite reaching the random forest model.

```
#50:50 model
batches = 5
epochs = 9
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,
                        layers=[1000, 500, 200],
                        metrics=accuracy,
                        wd = 20,
```

```
                        y_range = [0, 1])
learn.fit_one_cycle(epochs, lr_max=slice(1e-6,1e-4))
```

This model was easier to train, and it is interesting to see that it achieves higher accuracy than the 50:50 deep learning model, although only slightly.

```
#80:20 model
batches = 10
epochs = 16

dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,
    layers=[1000, 400, 20],
    metrics=accuracy,
    wd = 0.1,
    y_range = (0, 1))

learn.fit_one_cycle(epochs, lr_max=slice(1e-7,1e-5))
```

Model	Accuracy	Precision	Recall	F - Measure
DL 50:50	0.672	0.686	0.634	0.659
DL 80:20	0.686	0.776	0.523	0.625

Table 4.2: Scoring of deep learning detention models.

4.1.2 Deficiencies

The hard part of predicting deficiencies is defining what is accurate. Given the exploration in chapter 3, we decided to use *MAE*, as given in equation 2.5. As explored in the data chapter, most inspections lead to zero deficiencies. Therefore a model predicting close to zero or the mean may seemingly perform well, although the result is nonsensical.

To validate that we did not encounter the abovementioned issue, we trained the model on the full dataset. Then we explored the accuracy of the predictions using both the full validation dataset and slices of it. This allows us to see how different kinds of vessels and subsets of the validation data impact the accuracy of the predictions and how generalizing the model is. Table 4.3 presents the results.

- $0 - \infty$ All validation data, the normal accuracy test.
- $0 - 5$ dataset as proposed in bucket target, comprises 87% of all vessels.
- $1 - \infty$ dataset excluding inspections leading to zero detentions.
- $6 - \infty$ dataset as proposed in bucket target, 13% of vessels.
- *MAE* per target number of deficiencies.

Random Forest (RQ1)

```
m = RandomForestRegressor(n_jobs=-1,
                          n_estimators = 800,
                          max_samples = 4000,
                          max_features = 0.5,
                          min_samples_leaf = 3,
                          oob_score = True)
```

Bucket	$0-\infty$	$0-5$	$1-\infty$	$6-\infty$
RF	1.59	2.25	2.63	6.45

Table 4.3: MAE scoring of random forest regression models per bucket.

As discussed in chapter 3, the data chapter, the more deficiencies found in an inspection, the more of an outlier it is. Therefore, we should see the accuracy decrease. The number of samples also decreases when we go higher, which may become an issue.

Table 4.4 lists the number of samples available per number of deficiencies. Figure 4.1 shows the results.

In line with what we thought, the model starts to diverge when moving from the mean of 1.89 found in chapter 3.3, the data exploration chapter. The result of this is that the model has a hard time generalizing, especially for the outliers, capturing that most vessels are in reasonably good condition. The line also diverges on the conservative side, starting from about seven deficiencies, corresponding to when the sample sizes decrease significantly. Table 4.3 captures the same behavior.

Deficiencies	0	1	2	3	4	5	6	7	8	9
Sample Count	4492	1104	960	732	509	380	305	212	140	132

Deficiencies	10	11	12	13	14	15	16	17	18	19
Sample Count	96	68	63	37	39	26	27	10	11	10

Table 4.4: Validation samples available per target number of deficiencies.

From Figure 4.2, we can see that the statistical trend of the true values is captured. On the sample level, this is not as pronounced, given by the true values and predicted values not lining up. This also matches Figure 4.1, where the *MAE* increases further from the mean, showing that the model is conservative for the outliers.

Figure 4.3 shows the number of predicted deficiencies per vessel type. Again the statistical findings from previous research are shown with general cargo having the most deficiencies.

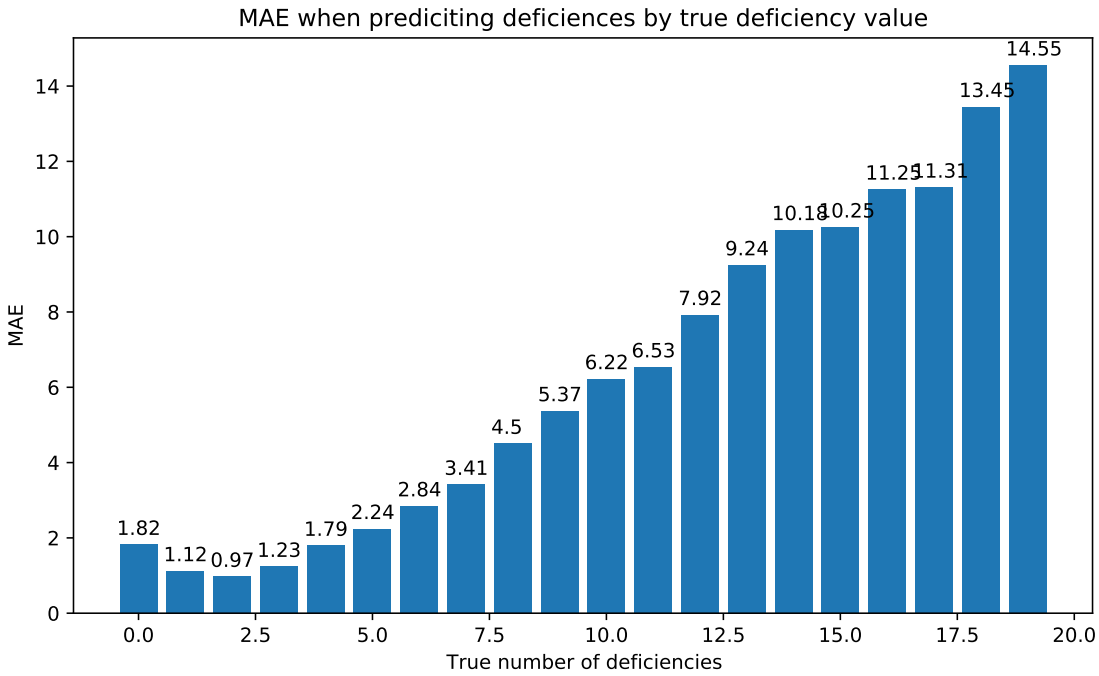

Figure 4.1: MAE when predicting different numbers of deficiencies using the simple dataset and random forest model.

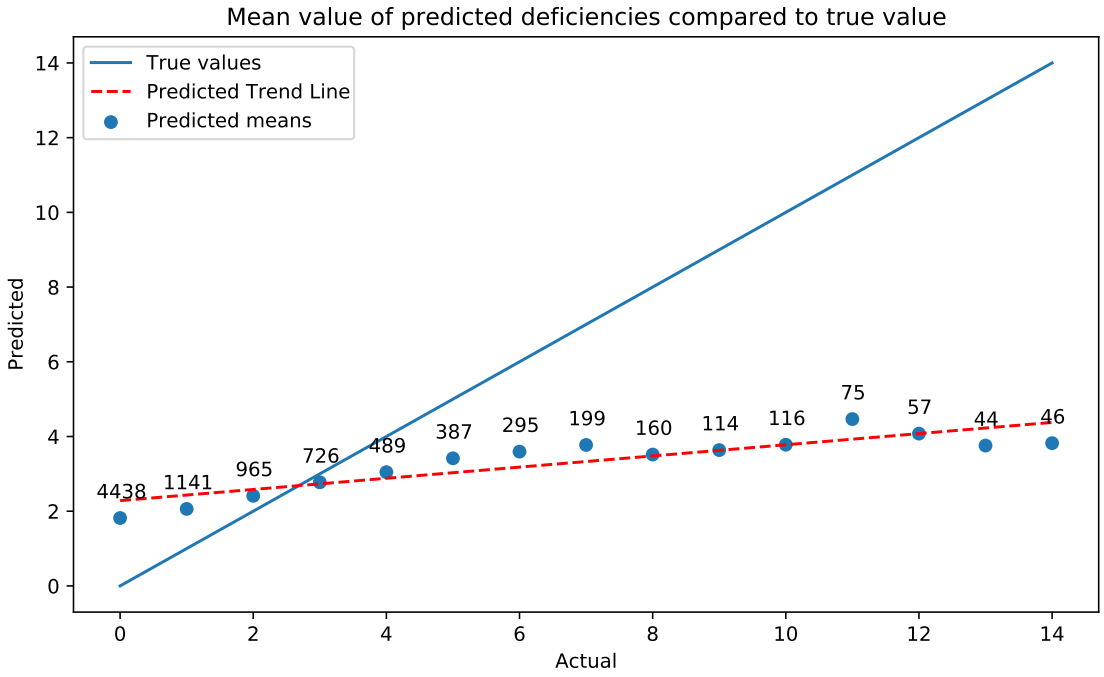

Figure 4.2: Mean predicted value per actual value for simple random forest regression model. Blue line would be perfect a statistical match. Number of samples leading to the mean listed as annotations. I.e. the model does not perform very well.

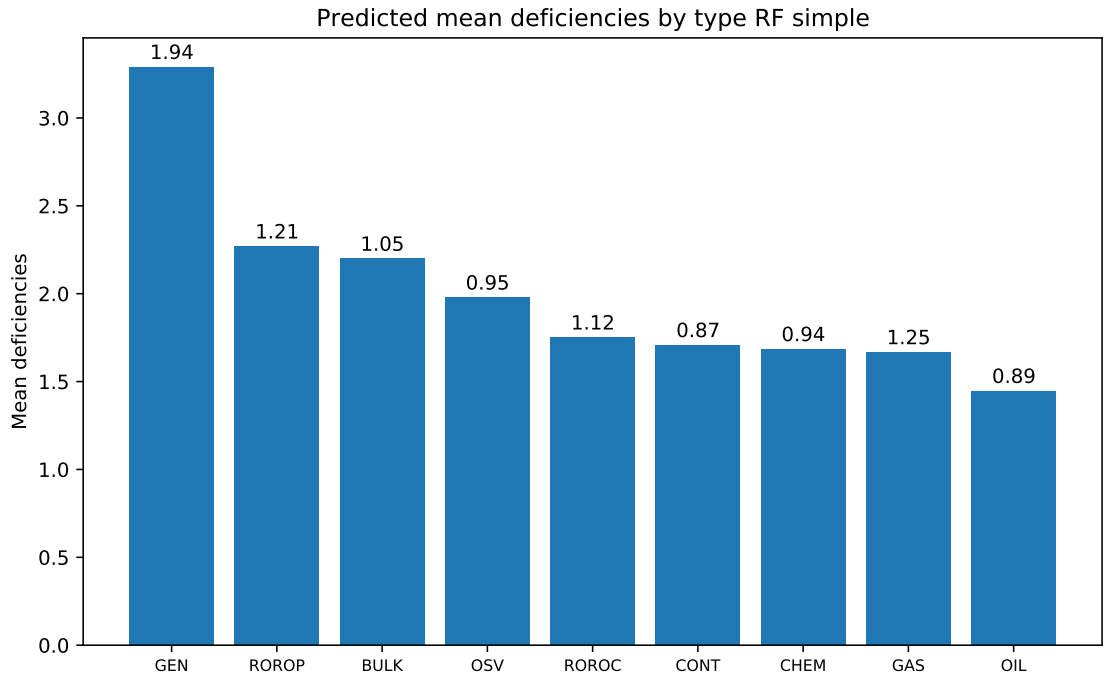

Figure 4.3: Predicted mean number of deficiencies by ship type. Standard deviation of the type annotated above each bar. For explanation of abbreviations see Table 3.6.

Deep learning (RQ2)

We created the deep learning model using *fast.ai* and the tabular learner core. Using a network with three hidden layers of 5000, 1000, and 200 nodes. Both `MSE` and `SmoothL1Loss`[1] as loss functions were explored. MSE is harder against outliers but does not work as well with high numerical values and may have issues with exploding gradients. Table 4.5 contains the results.

The `MSE` model would start to overfit the model towards the lower values. The *MAE* of all and those targeting more common values would decrease while the $6 - \infty$ set would increase. Since this is contrary to our goals, we decided to stop the training when that happened. The implementation is listed below.

```
# MSE Implementation
batches = 20
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls, layers=[2000, 1000, 200],
                        metrics=accuracy,
                        wd = 0.01,
                        y_range = (y_min, y_max),
                        loss_func=F.mse_loss
                       )

learn.fit_one_cycle(3, lr_max=slice(1e-6,1e-4))
```

[1] https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html accessed 2021-02-17

```
print_validation(learn)
learn.fit_one_cycle(5, lr_max=slice(1e-6,1e-4))
print_validation(learn)
```

Similarly to the `MSE` model, the `SmoothL1Loss` would also start to overfit towards the lower end of the spectrum leading. The box below contains the implementation.

```
# SmoothL1Loss implementation
batches = 80
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls, layers=[5000, 1000, 500],
                        metrics=accuracy,
                        wd = 0.1,
                        y_range = (y_min, y_max),
                        loss_func=torch.nn.SmoothL1Loss(),
                       )
learn.fit_one_cycle(5, lr_max=1e-3)
learn.fit_one_cycle(5, lr_max=1e-3)
learn.fit_one_cycle(5, lr_max=1e-3)
```

Table 4.5 shows the deep learning regression model results on the simple dataset, confirming that `MSE` is harsh to outliers. The model is forced to converge more to include them, leading to better accuracy. The deep learning model achieves similar accuracy compared to the random forest model. The *MAE* per deficiency for the `MSE` model is presented in Figure 4.4.

Model	$0-\infty$	$0-5$	$1-\infty$	$6-\infty$
DL *MSE*	2.41	1.83	2.71	6.17
DL *SmoothL1Loss*	1.98	1.22	2.95	6.88

Table 4.5: MAE scoring of deep learning regression models.

4.2 Engineered dataset

As described in the data chapter, chapter 3, we created the engineered dataset. Since this is the most advanced dataset, although also simpler in some regards due to the complexity of the dataset as described in the process of creating it, we implemented most models using this dataset. An important consideration is the differing sizes of our datasets. The engineered dataset with a time lag of 3 gives 34,966 samples, compared to the simple dataset, which has 62,828. Each sample has 4828 variables, leading to 5261 when processed.

4.2.1 Detentions

Again, the same 50:50 splits and 80:20 splits, as described above, were implemented with the same validation percentage of 30%.

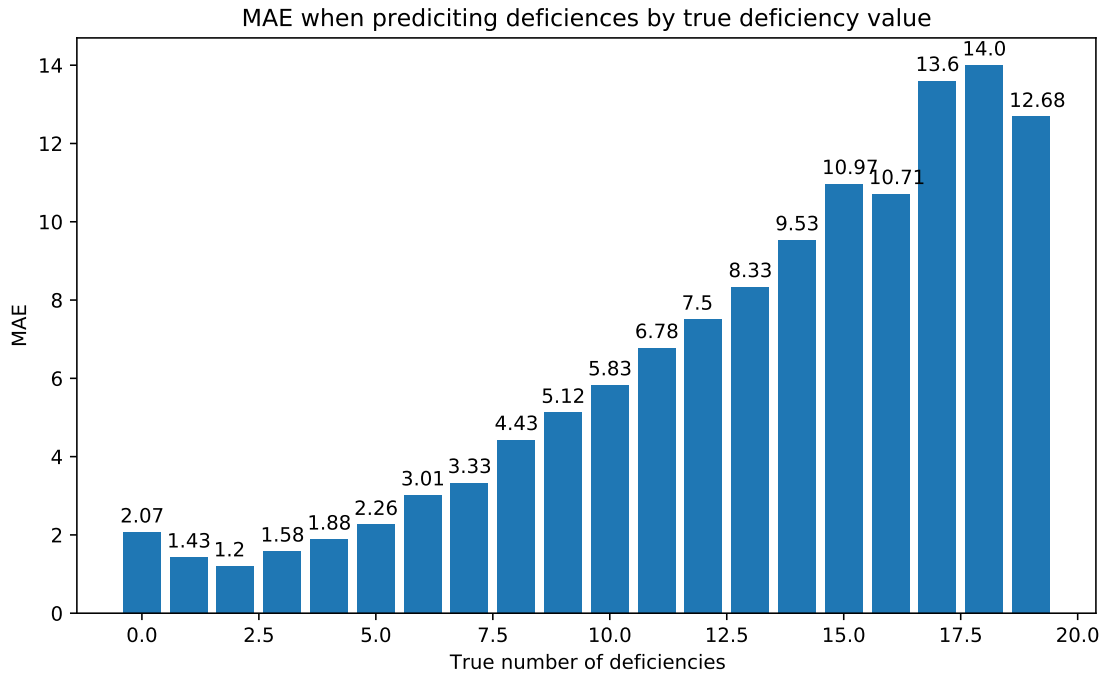

Figure 4.4: Mean absolute error (MAE) when predicting deficiencies using the simple dataset and deep learning `MSE` model.The `MSE` deep learning model has a hard time catching the outliers.

Random Forest (RQ1)

The same model parameters used for the simple dataset generalized well to the 50:50 random forest model using the engineered dataset. Due to the dataset's small size, there are no time constraints when training many trees. The 50:50 random forest model is based on 1788 train samples and 766 validation samples.

Table 4.6 contains the results. The interesting thing is that we achieve similar results to the simple dataset. We visualized some of the 4800 trees to validate that no data leakage or other mistakes had occurred. Figure 4.6 visualizes one of the 4800 decision trees used in the model.

```
m = RandomForestRegressor(n_jobs=-1,
                          n_estimators = 4800,
                          max_samples = 200,
                          max_features = 0.5,
                          min_samples_leaf = 0.0025,
                          oob_score = True)
```

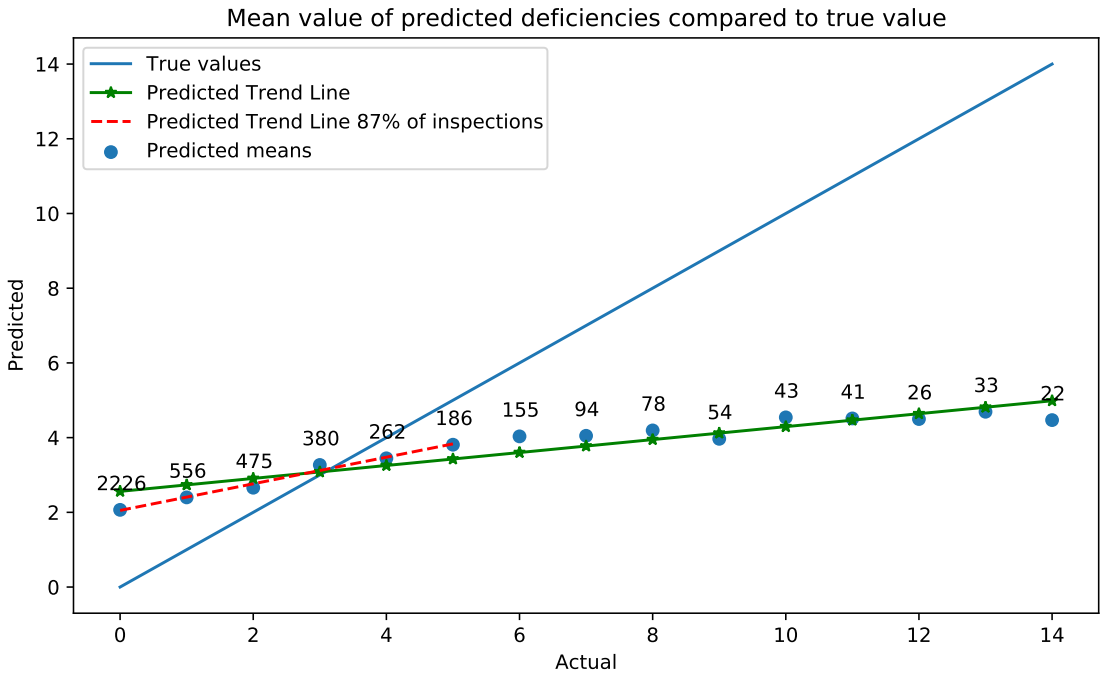

Figure 4.5: Mean predicted value per actual value for simple `MSE` DL model. Blue line would be perfect a statistical match, red line is 87% of the inspections. Number of samples leading to the mean listed as annotations.

Model	Accuracy	Precision	Recall	F - Measure
RF 50:50	0.705	0.725	0.661	0.691
RF 80:20	0.523	1.000	0.047	0.090

Table 4.6: Scoring of engineered random forest detention models.

Deep learning (RQ2)

The deep learning model was trained on 1788 samples and validated using 766 samples for the engineered sample. Table 4.7 contains the accuracy results. The interesting takeaway is that we reached the same accuracy even though we have both limited and expanded our dataset.

```
# 50:50 model
batches = 5
epochs = 9
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,layers=[1000, 500, 200],
                        metrics=accuracy,
                        wd = 0.1,
                        y_range = [0, 1])

learn.fit_one_cycle(epochs, lr_max=slice(1e-6,1e-4))
learn.fit_one_cycle(5, lr_max=slice(1e-6,1e-4))
```

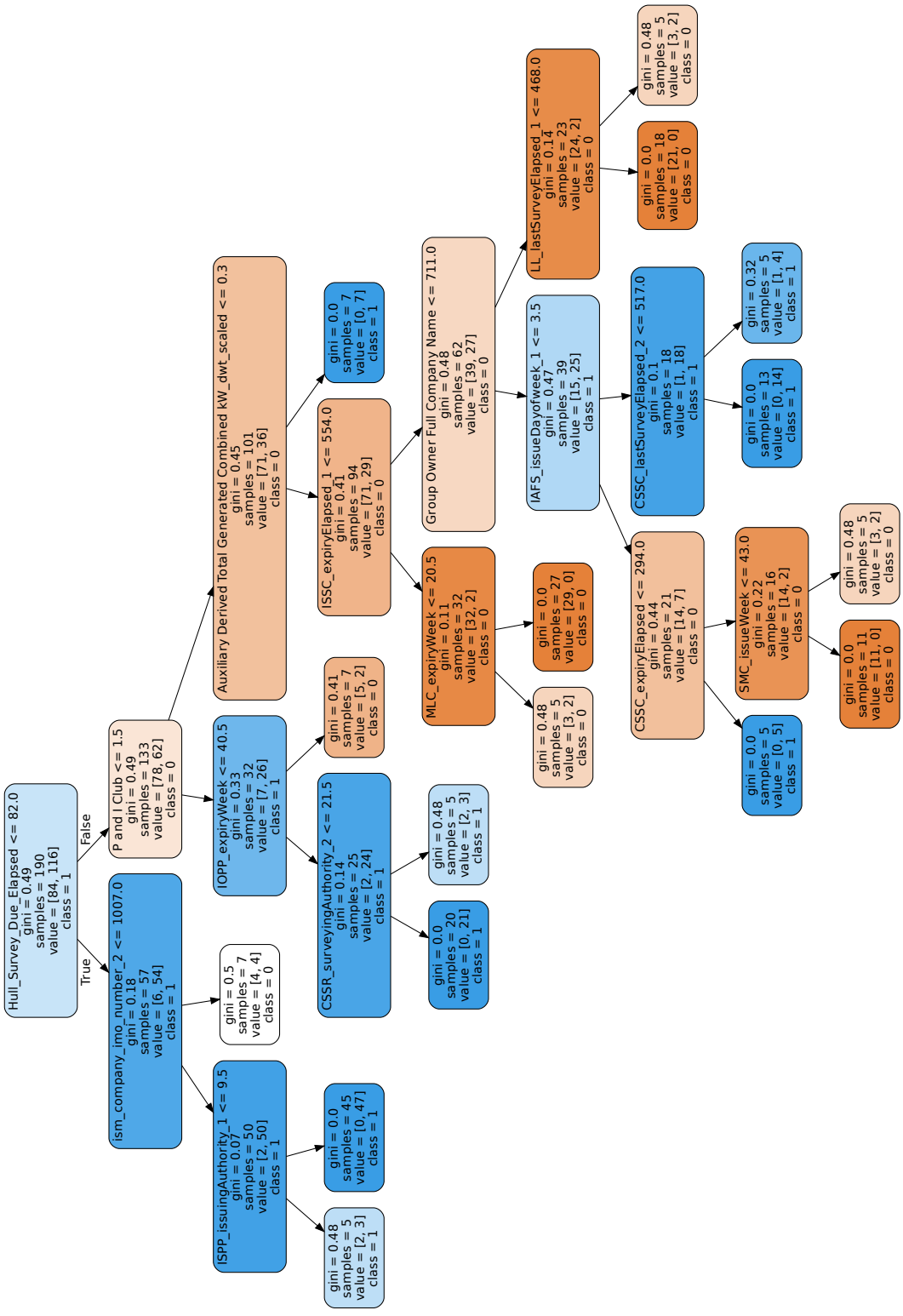

Figure 4.6: One of the trees generated for the engineered 50:50 split dataset.

As with the other 80:20 model, it converged, but we had a hard time improving the results by tuning the hyperparameters. Some other more balanced split could be better, or weighting of the output classes. We leave that as future work.

```
# 80:20
batches = 5
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,layers=[500, 250],
                        metrics=accuracy,
                        wd = 0.01,
                        y_range = [0, 1])

learn.fit_one_cycle(5, lr_max=slice(1e-6,1e-4))
learn.fit_one_cycle(5, lr_max=slice(1e-6,1e-4))
```

Model	Accuracy	Precision	Recall	F - Measure
DL 50:50	0.691	0.689	0.695	0.692
DL 80:20	0.685	0.710	0.627	0.666

Table 4.7: Scoring of deep learning model on the engineered dataset.

FCN and Resnet (RQ2)

As described by Fawaz et al. (2019), we applied the FCN and Resnet deep learning models to the detention classification dataset. Since the detention dataset is a binary classification, we must change the loss and activation functions to their binary counterparts. Thus we changed the loss function from *categorical_crossentropy* to *tf.keras.losses.BinaryCrossentropy()* and activation of the output layer from `softmax` to `sigmoid`. Otherwise, they were left as is.

We decided to train the `FCN` and Resnet models on the 50:50 dataset since it had already proven to achieve similar results as the 80:20 dataset but with easier hyperparameter tuning.

For the `FCN` model, the highest accuracy achieved over 200 epochs was 0.590. Since the accuracy would need to be higher to be of interest to the other models, we decided not to tune the model further.

The Resnet model performed better, reaching an accuracy of 0.65. However, since that still does not come close to other models, we decided not to tune the model's hyperparameters further.

4.2.2 Deficiencies

The results are in Table 4.8.

Random Forest (RQ1)

We created the random forest deficiencies model using the same 34,966 samples with a 30% validation split.

```
m = RandomForestRegressor(n_jobs=-1,
                          n_estimators = 3200,
                          max_samples = 5000,
                          max_features = 0.5,
                          min_samples_leaf = 20,
                          oob_score = True)
```

Model	$0-\infty$	$0-5$	$1-\infty$	$6-\infty$
RF	2.33	1.65	2.61	5.99
DL `MSE`	2.32	1.52	3.22	6.67
DL `SmoothL1Loss`	2.25	1.41	3.16	6.80

Table 4.8: MAE scoring of regression models on the engineered dataset.

Deep learning (RQ2)

`MSE`

```
batches = 40
epochs = 20

dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,
                        layers=[2000, 1000, 500],
                        metrics=accuracy,
                        wd = 0.01 ,
                        y_range = (y_min, y_max),
                        loss_func=F.mse_loss
                       )
learn.fit_one_cycle(5, lr_max=1e-2)
```

`SmothL1Loss`

```
batches = 80
epochs = 20
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,
                        layers=[5000, 1000, 500],
                        metrics=accuracy,
                        wd = 0.01 ,
                        y_range = (y_min, y_max),
                        loss_func=torch.nn.SmoothL1Loss(),
                       )
learn.fit_one_cycle(5, lr_max=1e-2)
```

4.2.3 Bucket target deficiencies

We drew conclusions from the detention data to implement the bucket target. The more even distribution, the better. Transforming the target to buckets allows predicting the number of deficiencies to become a classification task rather than a regression like we constructed earlier.

We used a stratified split based on the lowest percentage. Table 3.10 contains the distribution in the proposed splitting of the deficiency counts. With this bucketing, we can use $3 \cdot 12.9\% = 38.7\%$. For the validation, 20% of the samples were used.

This gives 11,404 samples in the train set and 2852 in the validation set.

Random Forest (RQ1)

The settings for the random forest model is given below.

```
m = RandomForestClassifier(n_jobs=-1,
                           n_estimators = 1600,
                           max_samples = 1000,
                           max_features = 0.5,
                           min_samples_leaf = 19,
                           oob_score = True)
```

Since we now have a multilabel classification, we can not use the Precision, Recall, or F-Measure numbers directly. They are for binary targets. Instead of visualizing the model performance, we break them out per class instead of doing some averaging or similar, which can also be used. From Table 4.13, we again see that we score over the random chance baseline of 1/3 but have a hard time getting better results.

Deep learning (RQ2)

Following previous results in the project, the neural network had a hard time outperforming the Random forest model. The network which performed the best was very simple, leading to an accuracy of 0.48.

```
batches = 12
epochs = 5
dls = to.dataloaders(bs=int(len(to.train.xs) / batches))
learn = tabular_learner(dls,
                        layers=[200, 100],
                        metrics=accuracy,
                        wd = 5,
                        y_range = (0, 1)
                       )

learn.fit_one_cycle(epochs, lr_max=slice(1e-6,1e-4))
```

FCN and Resnet (RQ2)

Since the target has three classes, the models were left as is, with the loss function being `categorical_crossentropy` and using `softmax` as the activation on the output layer.

Neither the FCN nor Resnet models converged in 50 epochs better than a random guess, giving an accuracy of 0.33, and time was running out. Therefore, we decided not to explore these models on the bucket target further.

4.3 Feature importance (RQ3)

Feature importance is of great importance when constructing machine learning models. Using feature importance, we can construct a final model only using the necessary features, leading to less noise and a smaller dataset. Feature importance allows one method for us to answer **RQ3**: How do the results of our constructed models relate to previous research?

An advantage of random forest models is their high interpretability: it is effortless to see which features are important when classifying data. CAM (Class Activation Mapping) achieves the same goal for the deep learning models, but that requires specific layouts of layers in a deep learning model. We explored feature importance for both the simple and engineered random forest models.

For the random forest models, we calculated two variations of feature importance: first, mean decrease in impurity, and after that, based on feature permutation. An issue with impurity-based feature importance is that it can be misleading on high cardinality features. We also calculated feature permutation-based importance to contrast this issue.

Interestingly we find several features not mentioned in previous research as the most impactful, for example, the date when the MLC (Maritime Labour Convention) certificate expires and the elapsed time since the last CSSC (Cargo Ship Safety Construction) certificate survey and the main engine model.

4.4 All results – Concise

In this section, all the results from the model exploration are collected and presented concisely. All detention models are listed in Table 4.11; all deficiency models are listed in Table 4.12; and all bucket models are listed in Table 4.13.

Feature	Importance	Description
number_deficiencies	0.08733	Previous inspection
Built Elapsed	0.06087	
P and I Club	0.02636	
Built Year	0.02508	
ism_company_imo_number	0.01640	
Hull_Survey_Due_Elapsed	0.01608	
Auxiliary Derived Total Generated Combined kW	0.01420	
Registered Owner Country/Region	0.01186	
Engine Derived Total Mechanical Generated (kW)	0.01137	
MLC_expiryElapsed	0.01011	Maritime Labour Convention
Last_Hull_Survey_Elapsed	0.00984	
port_name	0.00855	
TONN_issueElapsed	0.00832	International Tonnage Certificate
Main Engine 1 MCR (mkW)	0.00806	
Main Engine Model Series	0.00758	
Classification Society	0.00678	
CSSE_issueElapsed	0.00665	Cargo Ship Safety Equipment Certificate
Thrusters Total kW	0.00646	
ISSC_issueDayofyear	0.00614	International Ship Security Certificate
CSSC_issueElapsed	0.00614	Cargo Ship Safety Construction Certificate

Table 4.9: The most important features when calculating impurity-based feature importance for the simple random forest model with descriptions for the ones which are not self-explanatory.

Feature	Importance	Abbreviations
number_deficiencies	0.03245	Description
Registered Owner Country/Region	0.00851	
MLC_expiryYear	0.00651	Maritime Labour Convention
ism_company_imo_number	0.00598	
inspection_type	0.00569	
Built Year	0.0044	
CSSR_lastSurveyPlace	0.00429	Cargo Ship Safety Radio Certificate
Thrusters Total kW	0.00403	
BC01_expiryElapsed	0.00382	Bunker Oil Pollution Damage
Main Engine Model	0.00382	
DOC_issuingAuthority	0.00375	Document of Complicance
CSSC_lastSurveyElapsed	0.00372	Cargo Ship Safety Construction Certificate
IAPP_issueDay	0.00363	International Air Pollution Prevention Certificate
Manager Full Company Name	0.00352	
Operator	0.00349	
SMC_expiryDayofweek	0.00345	Safe Manning Certificate
DOC_lastSurveyElapsed	0.00343	Document of Complicance
Classification Society	0.00312	
Speed (knots)	0.00306	
Type	0.00302	

Table 4.10: The most important features when calculating permutation-based feature importance for the simple random forest model with descriptions for the ones which are not self-explanatory.

Dataset	Model	Split	Accuracy	Precision	Recall	F - Measure
Simple	RF	50:50	**0.722**	0.756	0.659	0.703
Simple	RF	80:20	0.522	0.892	0.051	0.096
Simple	DL	50:50	0.672	0.686	0.634	0.659
Simple	DL	80:20	0.686	0.776	0.523	0.625
Engineered	RF	50:50	0.705	0.725	0.661	0.691
Engineered	RF	80:20	0.523	1.000	0.047	0.090
Engineered	DL	50:50	**0.691**	0.689	0.695	0.692
Engineered	DL	80:20	0.685	0.710	0.627	0.666

Table 4.11: Scoring of all detention models.

Dataset	Model	Loss	$0-\infty$	$0-5$	$1-\infty$	$6-\infty$
Simple	RF	-	**1.59**	2.25	**2.63**	6.45
Simple	DL	`MSE`	2.41	1.83	2.71	6.17
Simple	DL	`SmoothL1Loss`	1.98	**1.22**	2.95	6.88
Engineered	RF	-	2.33	1.65	2.61	**5.99**
Engineered	DL	`MSE`	2.32	1.52	3.22	6.67
Engineered	DL	`SmoothL1Loss`	2.25	1.41	3.16	6.80

Table 4.12: MAE scoring of all regression deficiency models.

Model	Accuracy	Target bucket	Precision	Recall	F - Measure
RF	0.500	0	0.498	0.667	0.570
		$1-5$	0.399	0.285	0.333
		$6-\infty$	0.579	0.547	0.562
DL	0.483	0	0.481	0.69	0.567
		$1-5$	0.394	0.094	0.151
		$6-\infty$	0.502	0.666	0.572
FCN	0.33	Left out			
Resnet	0.33	Left out			

Table 4.13: Scoring of all the different bucket models.

Chapter 5
Discussion

5.1 Analysis of Data

As explored in the data chapter, the quality varies, with some parts being good and others missing much data. There should be much to be found in the dataset, though, and we can not think of any better sources than the ones used other than incorporating the Tokyo MoU PSC data.

Regarding the data cleaning performed, there is always more to do with a dataset centered around human input. For example, we easily fixed extreme outliers, but more subtle errors likely still exist.

Unreasonable outliers are easy to spot. The issue comes when outliers are in the normal range for the value in question. An example is engine power which ranges from hundreds to hundreds of thousands of kilowatts depending on vessel size.

To automatically find these issues, multivariate analyses must be applied. Doing reliably and consistently without large amounts of meticulous human labor is still an open question in the broader industry. For future work, we expect this to become easier as the current industry forming around data observability and quality becomes easy-to-use tools integrating into the data pipeline for machine learning projects. This applies both to us as data consumers and the platforms providing us with the data.

There are also significant issues with the encoding used for the model. Due to memory constraints, we had to resort to first label encoding and then code counting, losing much data.

Especially label encoding is egregious. An example would be encoding ten classification societies labeled from 0 to 9. The random forest model then creates a split in this where it gains the most information. The issue starts if there are two bad classification societies on both sides of the split, vastly lowering the information gain. One hot encoding would be better here since it would allow the selection to be precise, although that carries its problems with overfitting.

5.2 Analysis of Final Models

In all models, we find a signal with the same characteristics as from previous statistical work. The hard part is bringing this down to the sample level with satisfactory accuracy, which has been very hard. Generally speaking, we conclude that we went too broad and incorporated too much data in the model leading to less satisfactory results.

A consideration we should have explored for the engineered dataset is how the static and dynamic data interact in a time series manner. This is because the dataset has one static part and three dynamic parts. A consideration will be if the model becomes better and more regularized by changing this. For example, if the static data was repeated three times together with the dynamic, instead of keeping it as a separate chunk. This is only applicable to deep learning methods. For those, the position of the variables matters if, for example, a convolution is applied.

An interesting takeaway is that both the engineered and simple data performed similarly. We can see two bases for this result. Either the engineered dataset captures more information, but more is also thrown away due to the grouping and discarding done. The other possibility is that the classification mostly comes from the static part, which does not change.

Regarding the feature importance, unsurprisingly, the strongest factor is the previous state through the *number of deficiencies*. Similarly, other factors found through previous research are also found, like the company, region, and vessel type. More interesting are the fields that have yet to be found in previous research, for example, the time elapsed since the Bunker Oil Pollution Damage certificate was issued or where the Cargo Ship Radio Certificate last was surveyed. Interesting follow-ups would be to drill down deeper into these signals and explain why our models find them important.

5.3 Uncertainties

The most significant uncertainty is the quality of the data. It is all very variable and comes down to the question: Is there any information in a missing data point? E.g., if the owner only cares about updating the central registries with the bare necessity could correlate with how the vessel is operated. Finding which variables incorporate such ideas outside of pure noise would be hard.

Another uncertainty we have is forced selection due to the data available. Only the last four years of inspections are available online, which is what we used. This lends itself to both mean that the quality between years should be more consistent, but also makes longer trends harder to model.

5.4 Future Research and Improvements

Considering the methodology carried out and the results obtained from it, the following improvements could be made and are something to consider for future research:

- The data infrastructure needs to mature. In our experience using Pandas version 1.2.0, we had trouble when going above 5000 columns. This could have been mitigated by

keeping the data in lists or similar inside the columns, but then there are issues with writing a conversion to the data loader, and it is challenging to search and work with the dataset. Due to the size of the Paris MoU dataset, a regular computer was not sufficient. The work was shifted to remote work against a server with 192 GB of memory, which we accidentally crashed several times due to naïve algorithms consuming too much memory. Another method may be to try the methods used, e.g., images, encoding the data into files, using the folders for data-loading, or loading all data into a commercial data warehouse. This would likely also allow for One Hot Encoding and similar techniques for labeling the data, which we wanted to do, but could not due to infrastructure limitations.

- We briefly explored principal component analysis (PCA), but it resulted in no material improvement in accuracy, although it resulted in a less complex dataset. A significant issue with PCA is that much of the interpretability is lost when combining fields to reduce the dimensionality. Further investigation is needed to conclude the complete applicability of PCA on this dataset.

- Consistent encoding of the categorical labels is another topic we decided not to explore due to time constraints. We applied the default implementation, which starts from the top. The default is a good baseline, but since we had columns that could have corresponding data, e.g., issuing authority of certificates, something akin to a global hash table could help create more correlations within the dataset.

- Vabalas et al. (2019) describe three sources in which a larger feature set leads to the model learning more statistical noise. The referenced articles are old, long before the advent of deep learning. Follow-up studies should be more rigorous in selecting features to maximize impact instead of taking the broad approach detailed here. Start small and add features instead of relying on the algorithm to sort out the ones that do not contribute to the classification/regression.

- There could have been more feature engineering since many fields depend almost entirely on vessel size. There is also an entire can of worms considering the emissions data and ship types. This creates numbers that are very hard to compare. Then comes another can of worms when adding passengers or more complex ship operations to the equation.

- Another foray into this could be leaning more into the anomaly detection field, which includes isolation forests and heavier penalties for wrong classifications.

- Another method we considered but have yet to try would be creating clusters of vessels and a model for each cluster. These clusters are then baked into a larger model. An easy split would be the ship types; others could be by size or, say, the main engine manufacturer. The only need is to keep the data size large enough. Clustering greatly increases the work that needs to be done, but much of the infrastructure to find the right features could likely be reused.

- Another dataset that could be incorporated is the other MoU regimes, especially Tokyo's MoU, which previous research has found to have a very high quality of data.

- With the methods we used, we have demonstrated that random forests perform as well as neural networks. A way to further increase the performance is to use, e.g., boosting methods where ensembles of models are combined.
- As said in the theory chapter, our deep learning model choices would have allowed the use of CAM, which we decided not to explore due to time constraints. A method to go forward and contrast our result would be to use the random forest trees and CAM to see where they correlate and, most importantly, do not correlate and try to understand why that happens and engineer features based on this.
- There is also the question of whether an automatic method could be applied to the data cleaning to find the unreasonable data points. The dataset is large enough to make a complete look-through as a human infeasible, but quality should be increased if a smaller number could be flagged.

Besides the improvements, the dataset could be used for the following analyses:

- Further, explore correlations between environmental data and safety or other static variables by isolating parts of the entire source dataset while allowing targets in the MoU dataset.
- Connect it to the EMCIP and Clarksons Research Incidents dataset and broaden the exploration from PSC also to incorporate incidents. This may yield different and interesting contrasting results.
- The incident data could be linked to environmental concerns.
- Explore the possibility of predicting which deficiency groups a vessel has. I.e., if we can find correlations between fire safety and some characteristic on a sample level.
- There could be exploration done regarding feature engineering of the deficiency data leading to a more dense, easily applied structure. Some middle ground should be able to be found, which also captures the deficiency areas and subareas. In this thesis, we went to both extremes, incorporating all or grouping everything.

Chapter 6
Conclusion

The thesis concludes that there is a signal in the data, confirming previous studies. However, it is hard to accurately bring this down from a statistical level to a sample level, which would allow us to predict the future state of individual vessels. Through the feature importance of the simple random forest model, we also found several fields that contribute to the model's final accuracy, which we have not seen mentioned in previous research, including the time since hull surveys and the MLC expiry date.

Regarding **RQ1** in which, we investigate the validity of the base model and the ability to predict a vessel's future state regarding its safe operations. The simple random forest model finds the same variation by type as previous research, with the strongest being the simple random forest model predicting deficiencies. The mean deficiencies for the different ship types line up with previous research. Similarly to the previous risk indexes, we can also predict if the next inspection will lead to a detention with an accuracy of 72% together with a *F-Measure* of 0.703. Increasing the accuracy from the random forest model baseline proved more challenging than expected. However, with the considerations listed in Chapter 5, discussion, it should be possible to improve upon our results.

Comparing all models, the simple Random Forest model performed the best, although not to the level where the result is usable. The random forest models' accuracy could likely be increased by boosting or similar methods. The issue is that those methods incorporate more hyper-parameters leading to a more complex model to train. Another method to improve performance is to engineer the feature selection further to create more equally scaled variables. Regarding the deep learning model in **RQ2** for the detention dataset, they performed similarly to the random forest baseline when applying a split of 80:20 between the target classes. This split allows the use of more data and could be explored further. The accuracy of the deep learning models is similar but a bit worse compared to the random forest models. This result aligns with previous research applying deep learning to tabular data, showing once again that the data sets need to be very large for tabular deep learning models to shine.

The interesting takeaway is that we achieve similar results with less data for the engineered models both applied on **RQ1** and **RQ2**. Two outcomes could lead to this, which we

have not explored: either the data we removed was insignificant for the model, or the methods used captured enough information accurately with the larger time lag to create a similarly accurate model. Thus a good follow-up would be to engineer the dataset further to reduce the noise.

For the regression-based deficiency models constructed for **RQ1** and **RQ2**, we found that `MSE` loss performs better than `SmoothL1Loss`, especially for the outliers, e.g., $6 - \infty$ deficiencies, which is in line with the expectations.

Based on the results of the FCN and RESNET neural networks proposed by Fawaz et al. (2019), we conclude that the dataset is closer to a categorical dataset.

Regarding **RQ3**, the conclusion is that when drilling down into the models, investigating feature importance, and creating statistics regarding the samples used leading to the predictions, the same trends which have been found in previous statistical work are validated.

Ultimately, we created an entirely new dataset with a size unmatched by previous research. This dataset will be further studied and incorporated into other projects at RISE. We applied several machine learning and deep learning method to this dataset. The models provide a good hint but can not be considered conclusive in a real-world situation.

Therefore, the model results can be used as a baseline comparison for future research and give good suggestions of where to look deeper, both from a model and feature engineering perspective and the difficulty of the problem at hand. Feature engineering, or a large amount of systematic work to only select the features giving the most information, is required. We found the same signals as the statistical methods and beyond that high impact features never mentioned in previous research.

References

Abghari, S. and Kazemi, S. (2012). Open data for anomaly detection in maritime surveillance. Master's thesis, Blekinge Tekniska Högskola, School of Computing.

Baydogan, M. (2020). Multivariate time series classification data sets accessed 2021-11-20. *http://www.mustafabaydogan.com/*.

Bijwaard, G. E. and Knapp, S. (2009). Analysis of ship life cycles-the impact of economic cycles and ship inspections. *Marine Policy*, 33:350–369.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bye, R. J. and Aalberg, A. L. (2018). Maritime navigation accidents and risk indicators: An exploratory statistical analysis using ais data and accident reports. *Reliability Engineering and System Safety*, 176:174–186.

Cariou, P., Mejia, M. Q., and Wolff, F. C. (2006). On the effectiveness of port state control inspections. *Transportation Research Part E: Logistics and Transportation Review*, 44:491–503.

Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey.

Chollet, F. (2018). *Deep Learning with Python*. Manning.

Cruise Lines International Association (2019). Clia global passenger report 2018. Technical report, Cruise Lines International Association.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33:917–963.

Hancock, J. T. and Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, 7.

Hassel, M., Asbjørnslett, B. E., and Hole, L. P. (2011). Underreporting of maritime accidents to vessel accident databases. *Accident Analysis and Prevention*, 43:2053–2063.

Heij, C. and Knapp, S. (2015). Effects of wind strength and wave height on ship incident risk: Regional trends and seasonality. *Transportation Research Part D: Transport and Environment*, 37:29–39.

Hollnagel, E. (2018). *Safety-I and safety-II: the past and future of safety management*. CRC press.

Howard, J. and Gugger, S. (2020). *Deep Learning for Coders with fastai and PyTorch*. O'Reilly Media.

Hänninen, M., Banda, O. A. V., and Kujala, P. (2014). Bayesian network model of maritime safety management. *Expert Systems with Applications*, 41:7837–7846.

Hänninen, M. and Kujala, P. (2014). Bayesian network modeling of port state control inspection findings and ship accident involvement. *Expert Systems with Applications*, 41:1632–1646.

Knapp, S. and Franses, P. H. (2007). Econometric analysis on the effect of port state control inspections on the probability of casualty. can targeting of substandard ships for inspections be improved? *Marine Policy*, 31:550–563.

Li, K. X., Yin, J., Bang, H. S., Yang, Z., and Wang, J. (2014a). Bayesian network with quantitative input for maritime risk analysis. *Transportmetrica A: Transport Science*, 10:89–118.

Li, K. X., Yin, J., and Fan, L. (2014b). Ship safety index. *Transportation Research Part A: Policy and Practice*, 66:75–87.

Piniella, F., Az, E. R. G.-D., and Alcaide, J. I. (2014). A comparative analysis of vessels detained under the psc agreements of paris, tokyo and viña del mar. *Journal of Shipping and Ocean Engineering*, 4:291–306.

Psarros, G., Skjong, R., and Eide, M. S. (2010). Under-reporting of maritime accidents. *Accident Analysis and Prevention*, 42:619–625.

Tsou, M. C. (2019). Big data analysis of port state control ship detention database. *Journal of Marine Engineering and Technology*, 18:113–121.

UNCTAD (2019). Unctad handbook of statistics 2019. Technical report, UNCTAD.

Vabalas, A., Gowen, E., Poliakoff, E., and Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLoS ONE*, 14.

Wang, S., Yan, R., and Qu, X. (2019). Development of a non-parametric classifier: Effective identification, algorithm, and applications in port state control for maritime transportation. *Transportation Research Part B: Methodological*, 128:129–157.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *Proceedings of the International Joint Conference on Neural Networks*, 2017-May:1578–1585.

Wright, M. N. and König, I. R. (2019). Splitting on categorical predictors in random forests. *PeerJ*, 2019.

Xu, R., Lu, Q., Li, K., and Li, W. (2007). Web mining for improving risk assessment in port state control inspection. In *2007 International Conference on Natural Language Processing and Knowledge Engineering*, pages 427–434. IEEE.

Yang, Z., Yang, Z., and Yin, J. (2018). Realising advanced risk-based port state control inspection using data-driven bayesian networks. *Transportation Research Part A: Policy and Practice*, 110:38–56.

Appendices

INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2021-03-11

EXAMENSARBETE Investigating the Applicability of Deep Learning to Profile Ship Risk
STUDENT Mathias Kindberg
HANDLEDARE Pierre Nugues (LTH), Johannes Hüffmeier (RISE), Luis Sánchez-Heres (RISE)
EXAMINATOR Elin A. Topp (LTH)

Deep learning och förhindrandet av nästa kustnära oljekatastrof

POPULÄRVETENSKAPLIG SAMMANFATTNING **Mathias Kindberg**

Sjöfartsindustrin genomgår för närvarande en reformativ digital transformation som möjliggörs av data och global internettäckning. I detta arbete undersöker jag hur vi kan använda den datan för att förhindra nästa kustnära oljeutsläpp.

Deep learning är ett av de hetaste områdena idag. En störtflod av nya uppfinningar och tillämpningar dyker upp för var dag. Sjöfartsnäringen står idag inför en enorm utmaning att ta till sig dessa metoder och införliva de i användbara verktyg och ny kunskap.

Målet med detta arbete är att förutsäga fartygs framtida tillstånd. Där med möjliggöra ett till verktyg för att förhindra nästa kustoljeutsläpp. Detta görs genom skapa maskininlärningsmodeller baserade på data från hamnstatskontroller, fartygsegenskaper samt utsläppsdata.

Hamnstatskontroll är ett system utvecklat på 1980-talet för att bekämpa uppkomsten av sjöfart av låg kvalitet. Systemet fungerar genom stickprovsinspektioner på utländska fartyg, där möjligheten finns att fartyget kvarhålls om de problem som hittats inte åtgärdas i tid. Således, om vi kan förutsäga om nästa inspektion skulle bli ett kvarhållande, kan vi mer effektivt rikta in oss på undermåliga fartyg och se till att de är säkra innan de lämnar våra hamnar.

När man tränar deep learning modeller behövs i allmänhet ett mål, exempelvis om en bild är en hund, som används för att skapa modellens parametrar. Detta gör att modellen kan generalisera utanför den data den tränades med, och därmed gör det möjligt att ta fram nya insikter. Det är detta som gör det möjligt att klassificera fartyg i framtiden. För att hitta den bästa metoden utforskades tre angreppssätt:

- Förutsäga om nästa inspektion kommer att bli ett kvarhållande.
- Förutsäga antalet brister vid nästa inspektion.
- Förutsäga antalet brister genom att gruppera resultatet.

Resultaten är att jag lyckas förutsäga om nästa inspektion kommer att leda till kvarhållande eller inte med en noggrannhet på 72,2%. Vilket talar för giltigheten av datan och modellen, men det är inte tillräckligt bra för att användas i verkligheten. När man utforskar modellens viktning av olika fartygsegenskaper visar sig flera nya egenskaper ranka högt som inte nämnts i tidigare forskning.

Exempel är: Tid till förfallodatum för skrovundersökning, tid till utgång av MLC-certifikatet, huvudmotor. Det skulle vara mycket intressant att se om framtida forskning kan koppla dessa till faktisk säkerhet eller om de helt enkelt är en annan proxy av till exempel fartygstyp eller storlek som länge visat sig vara statistiskt korrelerade med risk.