NTNU
Norwegian University of
Science and Technology

Data-Driven Analysis of Vessel Performance

Christian de Jonge

Master of Science in Engineering and ICT
Submission date: June 2017
Supervisor: Eilif Pedersen, IMT
Co-supervisor: Nicolas Lefebvre, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU – Trondheim
Norwegian University of
Science and Technology

MASTER'S THESIS IN MARINE CYBERNETICS
SPRING 2017

CHRISTIAN DE JONGE

Data-Driven Analysis of Vessel Performance

Objectives The objective is to investigate the vessel performance in relation to various sensory variables. An extra attention will be put into analyzing the performance loss over time to assess the hull-propeller performance. Hopefully this will make it possible to define a prognosis model in a context of maintenance operation. To analyze and assess the given data a data-driven approach will be used. Much attention will be payed to the preprocessing of the data. The data will be analyzed using statistical learning methods in both an unsupervised and a supervised framework.

Scope of the work The work to be done can be summarized by the following steps:

1. To perform a literature review on the problem and relevant methods.
2. To describe a set of methods that are relevant regarding data exploration:
   - Mathematical derivation,
   - Software implementation,
   - Definition of toy-cases to illustrate drawbacks and advantages of the methods.
3. To achieve a benchmark of programs and programming language that can be used for data analysis.
4. To preprocess the real data into a suitable dataset
5. To prepare a simulated dataset.
   - Necessary variables
   - Suitable time range
   - Enough data points
   - Preprocess the dataset
6. The developed approaches will be illustrated and validated on a simulated dataset before application to the real dataset.
7. To visualize and interpret the results.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of algorithms, simulations results, real data analysis, discussion and a conclusion including a proposal for further work. The thesis should be submitted within June 11.

Eilif Pedersen, IMT
Main supervisor

Nicolas Lefebvre, IMT
Co-supervisor

Preface

The work described in this master thesis is a part of the study program Engineering and ICT with a specialization within Marine Cybernetics at NTNU. This project thesis is carried out spring semester of 2017. With backgrounds both in ICT and Marine Technology this thesis is used to learn necessary theory and methods to extract and analyze information from marine related data.

The thesis is written for NTNU in cooperation with DNV-GL, who provided the data and relevant background information.

This thesis assumes the reader has basic knowledge of marine vessels and statistical analysis.

Acknowledgment

I would like to thank the people at DNV GL for consistent support throughout the process and for providing the data, necessary information and giving and understanding of the data and system at hand. I would especially like to thank Christos Chryssakis for continuously being involved in the progress of the thesis through Skype meetings, as well as taking his time to provide with necessary resources.

It has been a great pleasure to have Nicolas Lefebvre as my supervisor. He supported me through weekly meetings to discuss the work done and to how to proceed. The discussions with Lefebvre gave many good inputs and suggestions of which methods to use and how to interpret the results. I would also like to thank Eilif Pedersen for being my main supervisor. I would like to thank Erling Singstad Paulsen, who has been working with data from the same vessel as me. Our cooperation and many discussions throughout the semester has been of great value.

Finally I am deeply grateful for the support and encouragement received from friends and family.

Trondheim, 2017-06-11

Christian de Jonge

Summary and Conclusions

In this thesis the relation between vessel performance and various vessel and environmental variables were investigated using a data-driven approach. A total of 12 variables such as speed over ground and days since drydock were considered with data for almost three years. The performance loss of the vessel were calculated by measuring the vessel performance and comparing it to an expected performance, calculated by the use of computational fluid dynamics. The relation between performance loss and time were investigated in particular to assess the hull and propeller performance of the vessel. Statistical models were trained to predict the performance loss from the 12 variables. The models were analyzed to assess the relative importance of the different variables.

The relevant data were extracted and put on a suitable format. After this the data were preprocessed by the use of synchronization, variable redefinitions, outlier removal, mean centering and normalization. The prepared dataset were analyzed using principal component analysis to reveal structures in the unlabeled dataset, and to verify known relations.

Performance loss were simulated for three different cases. Several statistical learning methods as well as outlier removal were preformed on the simulated models. This was done to verify that the methodology would reveal the relationships in the simulated data and such that we could compare the simulated models with the real-world data. Both the linear and non-linear regression models were able to uncover the relationships in the simulated data, and improved the prediction error rate by as much as 86.8 % for the most complex simulated model.

The same methodology used on the simulated models were applied to the real-world data. A second degree polynomial regression model reduced the prediction error rate by 97.8 %, better then expected. The non-linear nearest-neighbor regression only reduced the prediction error rate by 66.2 %. The variables that were most important in the least-squares regression model were the variables related to the propulsion system of the vessel. When finding the best subset of variables, the propulsion variables were always present. The time variables where not able to reduce the prediction error rate significantly and it was impossible to draw any strong conclusions on the effect of time on the performance of the vessel. Thus, no prognosis model which can be utilized in maintenance could be made.

Sammendrag og Konklusjoner

I denne masteroppgaven ble forholdet mellom fartøyets ytelse og diverse fartøy- og miljøvariabler undersøkt ved hjelp av en datadrevet tilnærming. Totalt ble 12 variabler, som hastighet over bakken og dager siden tørrdokk, vurdert med data fra nesten tre år. Fartøyets ytelse ble beregnet ved å måle fartøyets ytelse og sammenligne det med en forventet ytelse, beregnet ved bruk av numerisk fluiddynamikk. Forholdet mellom ytelsestap og tid ble undersøkt ekstra nøye for å kunne vurdere fartøyets skrog- og propellytelse. Statistiske modeller ble trent til å forutsi ytelsestap fra de 12 variablene. Modellene ble analysert for å vurdere den relative betydningen av de forskjellige variablene.

De relevante dataene ble hentet ut og satt på et passende format. Etter dette ble dataene forhåndsbehandlet ved bruk av synkronisering, variable omdefinisjoner, fjerning av utenforliggere, gjennomsnittlig sentrering og normalisering. Det forberedte datasettet ble analysert ved bruk av prinsipal komponent analyse for å avsløre strukturer i datasettet og for å verifisere kjente relasjoner.

Ytelsestap ble simulert i tre forskjellige tilfeller. Flere statistiske læringsmetoder, samt fjerning av utenforliggere ble utført på de simulerte modellene. Dette ble gjort for å verifisere at metodene ville avsløre forholdene i de simulerte modellene, slik at vi kunne sammenligne de simulerte modellene med den virkelige dataen. Både de lineære og ikke-lineære regresjonsmodellene greide å avdekke forholdene i de simulerte modellene, og forbedret prediksjonsfeilraten med så mye som 86.8 % for den mest komplekse simulerte modellen.

Den samme metoden som ble brukt på de simulerte modellene, ble brukt på den virkelige dataen. En annengrads polynomisk regresjonsmodell reduserte prediksjonsfeilraten med 97.8 %, bedre enn forventet. Den ikke-lineære nærmeste nabo-regresjonen reduserte bare prediksjonsfeilraten med 66.2 %. Variablene som var viktigste i andregrads regressjonsmodellen var variablene knyttet til fremdriftssystemet til fartøyet. Når vi fant den beste delmengden av variabler var fremdriftsvariablene alltid tilstede. Tidsvariablene var ikke i stand til å redusere prediksjonsfeilraten vesentlig, og det var umulig å trekke noen sterke konklusjoner om effekten av tid på fartøyets ytelse. Dermed kunne ingen prognosemodell i forhold til vedlikehold lages.

Contents

List of Figures

List of Tables

Nomenclature

AIC Akaike Information Criterion.

AIS Automatic Identification System.

BF Beaufort Number.

BOG Boil-off Gas.

CFD Computational Fluid Dynamics.

CMA Centered Moving Average.

CSV Comma-seperated Values.

CV Cross-Validation.

GCU Gas Combustion Unit.

GPS Global Positioning System.

HDF Hierarchical Data Format.

HPP Hull and Propeller Performance.

IDE Interactive Development Environment.

kts Knots.

LNG Liquefied Natural Gas.

LV Latent Variables.

MRU Motion Reference Unit.

MSE Mean Squared Error.

NaN Not-a-Number.

NN Nearest-Neighbor.

Pandas Open source Python library for data handling and analysis.

PC Performance Value.

PCA Principal Component Analysis.

PL Performance Loss.

PSL Percentage Speed Loss.

Python High-level, general-purpose, interpreted programming language.

RSS Residual Sum of Squares.

SAS Statistical Analysis System.

SciKit-Learn Open source Python library for data mining and data anlysis.

SOG Speed Over Ground.

Spyder Open source IDE for Python.

STW Speed Through Water.

Notation

$\mathbb{R}^i$	An i-dimensional vector of real numbers
$\mathbb{R}^{m\times n}$	A m-by-n matrix of real numbers
$\mathbf{X}$	Dataset, consisting of observations $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]^T$
$\hat{\mathbf{X}}$	Bilinear subspace model of $\mathbf{X}$
$\mathbf{X}_c$	Mean centered dataset
$\mathbf{x}_i$	Observation i consisting of n variables, $\mathbf{x}_i \in \mathbb{R}^n$
$\mathbf{c}_i$	Variable i consisting of m observations, $\mathbf{c}_i \in \mathbb{R}^m$
$\mathbf{Y}$	Set of responses $[\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m]^T$
$\hat{\mathbf{Y}}$	Prediction of $\mathbf{Y}$
$\mathbf{Z}$	Score matrix, consisting of score vectors $[\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_m]^T$
$\mathbf{P}$	Loading matrix, consisting of loading vectors $[\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m]^T$
$\mathbf{E}$	Residual matrix
$\mathrm{E}[X]$	Expected value of a variable X
$\mathrm{Var}[X]$	Variance of a variable X
$\mathrm{Cov}[X, Y]$	Covariance between variables X and Y
R^2	Coefficient of determination, a measure of goodness of fit for a statistical model

$\mathscr{D}$	Proximity matrix
$d(\mathbf{a}, \mathbf{b})$	Distance metric between observation **a** and **b**
J	Objective function to be minimized in an optimization problem
$\epsilon \sim N(0, \sigma^2)$	ϵ is normally distributed with zero mean and σ^2 variance
t_Δ	Timedelta
t_r	Real time
t_v	Virtual time
P_d	Delivered power
P_s	Shaft power
R_T	Total resistance of vessel
V	Speed through water
P_d	Quasi-propulsive efficiency
R_{SW}	Still-water resistance
R_{AA}	Added resistance due to wind
R_{AW}	Added resistance due to waves
R_{AH}	Added resistance due to changes in hull and propeller condition
η_0	Open-water propeller efficiency
η_H	Hull efficiency
η_R	Relative rotative efficiency
Q	Shaft torque
n	Shaft speed
V_m	Measured speed
V_e	Expected speed

Δ	Displacement of vessel
A_C	Admirality coefficient
$\mathscr{T}$	Training data
β	Set of coefficients $[\beta_0, \beta_1, \ldots, \beta_n]^T$ in linear regression model
$\mathrm{Err}_{\mathscr{T}}$	Test error on independent test set
Err	Expected test error
$E[\overline{\mathrm{err}}]$	Expected training error
$\overline{\mathrm{err}}$	Training error
α	Tuning parameter for a model

Chapter 1

Introduction

Monitoring vessel performance has been an interest ever since the steam engine. Evaluation of vessel performance is getting increased attention of ship owners because the bunker expense is becoming an increasing part of the total service costs (Carlton, 2011). As the service costs are increasing, together with pollution and environmental aspects becoming a controversial topic in the marine industry, the urge to increase the vessel performance has never been higher. In IMO (2009) it is estimated that the shipping industry accounted for 3.3 % of the global CO_2 emissions in 2007. If ship owners are given a better understanding of the vessel performance they might care more and start working actively to reduce their footprint on the environment. IMO (2009) also states that more than 90 % of global trade is carried by sea and that this number is only expected to increase, and hence there will be more fuel used for shipping.

A great deal of fuel can be saved by optimizing the logistics of shipping operations. This can, for instance, be by optimizing speed, and/or optimizing the route with respect to weather. Another method to improve vessel performance is by optimizing the trim of the vessel, which is of increasing interest for ship owners and operators (Larsen et al., 2012).

The performance loss due to fouling varies significantly depending on the vessel, operational profile, anti-fouling measures, etc. Better knowledge of the fouling growth makes it possible to determine when a hull or propeller cleaning is economically beneficial. This information can be very useful for a ship operator.

Another trend in the marine industry is fuel saving methods and products that are being used in order to reduce the fuel consumption by a few percent. An improved method for detecting small changes in vessel performance would give customer and suppliers better confidence in how these methods and products perform. For example, anti-fouling paint manufacturers are eager to document their product in service conditions.

1.1 Background

The performance of a vessel in service is an expression of the power consumption to drive the vessel through the water at a given state (speed, loading, operational and environmental condition), relative to a previous state or a reference state.

Over the lifespan of a vessel, the power consumption is expected to increase as the vessel performance efficiency decreases. This means that the power consumption will increase for a certain speed or that the speed will be reduced for a given power consumption. This performance reduction is mainly due to fouling of the hull and propeller, but other attributes can also affect this performance, like corrosion or damages. While corrosion or damages can be nearly impossible to repair, fouling of the propeller and hull can be cleaned. However, the vessel is never fully returned to the same performance condition as the original state.

Using sensory data from an LNG tanker, the relationship between performance loss and various vessel and environmental parameters is explored. A particular attention will be paid to the performance loss over time, the hull-propeller performance. The data is available for a period of approximately three years with some periods of missing data. To investigate the relationship between performance loss and various parameters, statistical learning will be used in both an unsupervised and a supervised framework.

Hasselaar (2011) investigates how to develop an advanced vessel performance monitoring and analysis system. He highlights limitations related to monitoring and analysis of vessel performance, especially challenges related to the sensor system. There are many and good reasons to why one want to understand the behavior of the vessel in terms of ship power consumption and the speed of the vessel (power and speed loss in different loading and environmental conditions). Understanding this is useful in both economical and environmental aspects. Some of the benefits from understanding this behavior can be listed as

- Assessment of hull condition: When the speed of a vessel can be obtained with a set power at different stages of hull and propeller performance, the quality of any anti-fouling system can be assessed. The economically optimum interval for hull cleaning and dry-docking can be defined and economic delays due to fouling can be reduced by improved voyage planning.

- Assessment of engine condition: If we can measure a figure of engine efficiency, such as specific fuel consumption, the effects of any event occurring in the engine can be made visible. These events can, for example, be broken piston rings, fouled turbocharger, valve timing changes etc.

- Refinement of charter party agreements: When the capabilities and performance of a vessel can be determined irrespectively of environmental or loading conditions, agreements can be defined more precisely between ship owners and charter parties.

- Optimizing sailing performance: If the parameters that affect performance are monitored at frequent intervals, a large database becomes available which can be used to design an optimization system. Draft, trim, engine and autopilot settings in different operational and environmental conditions are some of the settings that can be optimized. Also, with frequent measurements of the vessel performance, the vessel's crew would be able to see the impact of their actions.

- Environmental assessment: As a response to the environmental global pressure, classification companies have introduced green certificates concerning ship pollution and efficiency. One example of this is DNV GL's 'clean' notation (DNVGL, 2017). To achieve a green certificate, quantity of emission gasses must be known. Newly built vessels perform their trials on the basis of calm waters and often without cargo. To find the service conditions, empirical correction factors are often used. The availability of a continuous performance monitoring system allows for assessment of emissions and helps in obtaining an environmental notation.

Pedersen (2014) uses artificial neural networks and Gaussian process regression on data from several vessels in combination with global atmospheric reanalysis data to analyze the vessel performance in terms of fuel consumption. He compares the data-driven methods to classical empirical methods and demonstrates how the data-driven methods can be used for evaluation of performance without any ship-specific information.

1.2 Previous Work

This thesis is a continuation of a project thesis carried out the fall semester of 2016. In this project thesis, sensory data from the vessel in consideration were preprocessed and analyzed using statistical methods from both an unsupervised and a supervised framework. The focus of the project thesis was to provide a necessary background for the methodology used in this thesis and to provide a solid understanding of the sensory data and vessel in consideration. In the project thesis, the vessel performance was not analyzed, as the focus was on the understanding, demonstration and visualization of the methods applied.

1.3 Objectives

The main objective of this Master's thesis is to analyze how the available sensory data affect the performance of our vessel, where extra attention will be put into analyzing the performance loss over time to assess the hull-propeller performance. The aim is then to define a prognosis model which can be utilized in maintenance. The main objectives of this Master's thesis are:

1. Evaluate which variables affect the performance loss and the importance of these variables.

2. Assess the predictive potential of chosen variables on the performance loss compared to an expected performance.

1.4 Limitations

There are several limitations to the work done in this thesis:

1. Sensory data from only one vessel is considered.

2. No information on the quality of the sensors is available, and sensory data is always contaminated by noise and faulty values to a certain extent.

3. The total period for which we have available data is limited to about three years, where the vessel is only dry-docking once.

4. The data we are handed are already preprocessed to some extent, how this is done is unknown.

5. Only a small selection of statistical methods are used.

Due to these limitations, the results in this thesis will at best be suggestive for the relation between selected variables and the vessel performance. We will still provide a solid foundation for how we one can proceed in further work.

1.5 Approach

To meet the objectives, different methods from the framework of statistical learning and statistical analysis will be utilized. The preprocessing methodology will be based on a set of predefined steps consisting of observation and variable selection, variable redefinitions, filtering and scaling. To assess the steps taken along the way, their strengths and limitations will be of high importance.

The data will be analyzed using statistical learning methods in both an unsupervised and a supervised framework, keeping in mind important limitations along the way. Extensive plotting and visualization of the data and results will be central to our approach. The approach can be summarized by the following steps:

1. The relevant data is extracted, preprocessed and analyzed in an unsupervised framework.

2. Virtual performance loss is simulated for three different cases.

3. The chosen statistical learning methods are performed on both the simulated and the real-world data, and the results are analyzed.

Using this approach will allow us to verify the applied methodology on simulated data before it is applied to the real-world data. It also allows for comparison of results between the simulated and the real-world data.

1.6 Structure of the Report

The rest of the report is organized as follows. Chapter 2 gives necessary information about the vessel, the available data and a brief understanding of the hull and propeller performance problem. In Chapter 3 an introduction to methods in statistical learning is given. Methods in both an unsupervised and a supervised framework will be covered, as well as relevant preprocessing. Model validation will also be discussed briefly. Chapter 4 will cover the database preparation in its entirety, from raw data to a synchronized, preprocessed dataset ready to be analyzed. In Chapter 5 we analyze the data using techniques from Chapter 2 and present our results for both the simulated and the real-world data. Chapter 6 will summarize and discuss our results and present some recommendations for further work.

Chapter 2

Vessel and System Description

The objective of this chapter is to provide necessary information about the vessel and describe the available data. This chapter also provides the theory on hull-propeller performance and the background assumption on which this theory is built. Section 2.1 gives a brief description of the vessel and how the vessel operates. In Section 2.2 the hull-propeller performance problems are discussed. Section 2.3 goes into detail about the available data and in Section 2.4 the software platform used in this thesis is discussed.

2.1 Vessel description

The vessel in consideration is a 300-meter long Liquefied Natural Gas (LNG) carrier designed to transport the LNG over long distances. Some common measurements of the vessel are presented in Table 2.1.

Name	**Parameter**	**Value [unit]**
Overall length	L_{OA}	295.0 m
Perpendicular length	L_{PP}	284.0 m
Breadth	B_M	43.4 m
Depth	D_M	26.0 m
Design draft	d_D	11.5 m
Transverse projected area	A_t	1547.3 m^2

Table 2.1: Common measurements of the vessel.

The vessel is equipped with four dual fuel generators from Wärtsilä, four cargo tanks, and has a twin screw propulsion system. Two of the generators has an output of 11000 kW and the other two has an output of 5500 kW, giving the vessel a maximum of 33000 kW. The engines can run either on natural gas, light fuel oil or heavy fuel oil and are designed to provide the same output regardless of the fuel. This allows the engines to use the excessive boil-off gas (BOG) from the LNG-tanks for propulsion.

When the ship operates at low speeds it is not able to use all the BOG for propulsion and the remaining BOG is handled by a gas combustion unit (GCU). The GCU burns the excessive BOG and releases the by-products into the atmosphere. For this particular vessel, during a laden voyage, as much as 4 tons of BOG is burnt and released into the atmosphere every hour. In cases where the need for propulsion exceeds that of available BOG, LNG can be taken from the tanks, often referred to as forced boil-off gas.

The paint used on the hull is a self-polishing paint delivered by Jotun. The self-polishing effect means that the hull efficiency will slightly increase for some time after a repaint before the efficiency starts to decrease Jotun (2017).

2.2 Hull and Propeller Performance Overview

Hull and propeller performance (HPP) refers to the relationship between the condition of a vessel's underwater hull and propeller and the power required to move the vessel through water at a given speed. Measurements of changes in vessel specific HPP over time makes it possible to indicate the impact of hull and propeller maintenance, repair and retrofit activities on the overall energy efficiency of the vessel in question. The decrease we see in HPP over time is mainly caused by fouling which is a general term to describe marine growth that attaches to a vessel. Biologically the fouling can be divided into micro-fouling (algae attachments such as "slime") and macro-fouling (barnacles and seaweed) (Callow and Callow, 2002). Fouling start to develop the moment an object is immersed in water. According to a study done by Eniram (2012), there are many parameters that influence the fouling process. These can be seen in Figure 2.1.

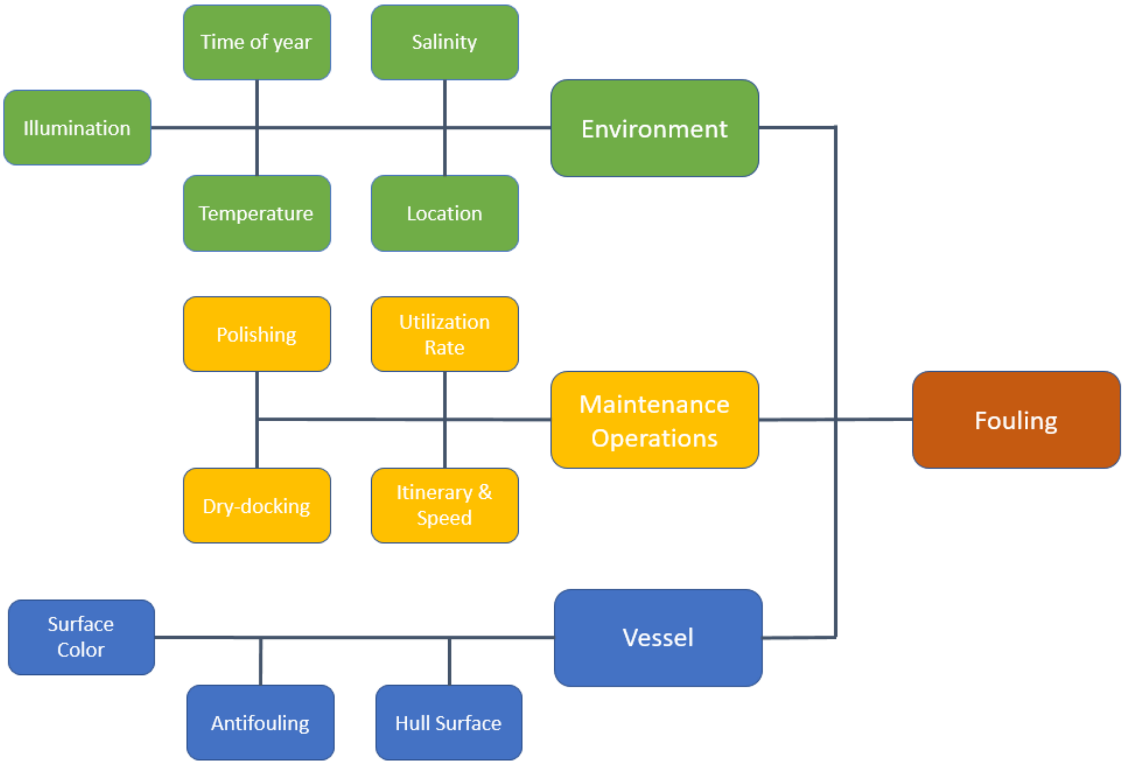


Figure 2.1: Parameters that effect the fouling process divided into three categories

Based on MARINTEK experience, the hull fouling of tank ships typically results in speed reductions of 5 % between dockings, corresponding to a power increase of 15 % and an increase in frictional resistance of 20 %. By increasing the docking frequency, the average loss could be reduced, resulting in a net power saving of about 5 % (IMO, 2009).

According to ISO (2015), the hull and propeller performance is closely linked to the vessel performance and vessel resistance. The performance of the vessel can be modeled based on the relation between the delivered shaft power and the total resistance where the delivered shaft power, P_d, can be expressed as

$$P_d = \frac{R_T \times V}{\eta_Q} \tag{2.1}$$

where R_T is the total resistance of the vessel, V is the vessel speed through water and n_Q is the quasi-propulsive efficiency. The total resistance consists of multiple components and can be written as

$$R_T = R_{SW} + R_{AA} + R_{AW} + R_{AH} \tag{2.2}$$

where R_{SW} is the still-water resistance, R_{AA} is the added resistance due to wind, R_{AW} is the added resistance due to waves and R_{AH} is the added resistance due to changes in hull and propeller condition (fouling, mechanical damages, bulging, paint film blistering, paint detachment etc.). Likewise, the quasi-propulsive efficiency consists of different efficiency components

$$\eta_Q = \eta_0 \eta_H \eta_R \tag{2.3}$$

where η_0 is the open-water propeller efficiency, η_H is the hull efficiency and η_R is the relative rotative efficiency. From this we can then express the hull and propeller added resistance as

$$R_{AH} = \frac{P_D \times \eta_Q}{V} - (R_{SW} + R_{AA} + R_{AW}) \tag{2.4}$$

The vessel speed through water, V, can be measured while delivered power, P_D, must be approximated. One way to do this is through calculations of the shaft power, P_S, by measuring the shaft torque and shaft revolutions as seen below:

$$P_S = \frac{2\pi}{60}(Q_s n_s + Q_p n_p) \tag{2.5}$$

where Q is the torque [kNm] and n is the shaft speed [min^{-1}]. The subscripts are indicating starboard or port as we have a twinscrew vessel.

For a vessel in service, both environmental conditions and operational profile (e.g. speed, loading, trim) vary. In order to measure changes in the speed-power relation for a vessel in service, one must compare two periods (a reference period and an evaluation period) where the environmental conditions and the operational profile are adequately comparable (filter the observed data) and/or apply corrections (normalize the observed data).

If we do not have measurements of certain variables, they can be estimated through various methods. These methods introduce additional uncertainty.

Measurements of ship specific changes in hull and propeller performance can be used in a number of relevant performance indicators to determine the effectiveness of hull and propeller maintenance, repair and retrofit activities. In Table 2.2 you can see 4 different performance indicators and their definition, as defined by ISO (2015).

Performance Indicator	Definition
Dry-docking performance: Determining the effectiveness of the dry-docking (repair and/or retrofit activities)	Change in hull and propeller performance following present out-docking (Evaluation period) as compared with the average from previous outdockings (Reference periods).
In-service performance: Determine the effectiveness of the underwater hull and propeller solution (including any maintenance activities that have occurred over the course of the full dry-docking interval)	Average change in hull and propeller performance from a period following out-docking (Reference period) to the end of dry-docking interval (Evaluation period).
Maintenance trigger: Trigger underwater hull and propeller maintenance, including propeller and/or hull inspection	Change in hull and propeller performance from the start of the dry-docking interval (Reference period) to a moving average at a given point in time (Evaluation period)
Maintenance effect: Determine the effectiveness of a specific maintenance event, including any propeller and/or hull cleaning	Change in hull and propeller performance from before (Reference period) to after a maintenance event (Evaluation period).

The four leading sources of uncertainty in the performance indicator are

- **Model Uncertainty**
- **Human Error**
- **Instrumental Uncertainty**
- **Sampling Error**

If there are no other losses than HPP, changes in vessel performance are fully due to fouling of the hull and propeller. The change in the vessel performance is now called the performance loss (PL). The PL is defined as the percentage loss of speed between a measured value V_m and an expected speed V_e for a given power consumption.

$$\text{PL} = 100\frac{V_e - V_m}{V_e} \tag{2.6}$$

Using this formula, positive values implies worse performance than expected. The expected speed at a given power consumption can be calculated using:

1. Computational fluid dynamics (CFD).
2. Found when the vessel is sailing in calm conditions (little wind and waves) before any significant fouling has taken place.
3. By scaling model experiments using empirical formulas.

In ISO (2015), a method for calculating the PL of a vessel is suggested. This method includes relevant sensors, minimum logging frequencies for these sensors, filtering methods and necessary calculation and assumptions. One of the main assumption for this methods is that the vessel speed, preferably speed through water, is logged no less than once every 15 seconds. This is no way close to the average logging frequency for our data of once every third hour. Hence I cannot follow this method to calculate the PL.

Ideally, we should use speed through water as both the measured and expected speed of the vessel when calculating the performance loss (Hasselaar, 2011). Speed over ground can be used as a substitute if we assume little current and no sideways drifting during turns.

Normally we see quite a lot of scatter in the performance loss of the vessel, there are several reasons why this might be, some of them are listed below:

1. Leeway drift is the drift caused by the component of the wind vector that is perpendicular to the object's forward motion.
2. Changes in boundary layer which may be caused by
   - Speed
   - Draft and Trim
   - Hull fouling
3. Stratified current layers
4. Excessive ship motion (mainly pitch)

One example of how the performance loss might develop over time can be seen in Figure 2.2 as assumed by Gundermann and Dirksen (2016). Here they assume that under normal conditions and when no husbandry actions are taking place, the level of the added resistance (R_{AH}) develops as the second part of an S-shaped growth curve as seen in Figure 2.2.

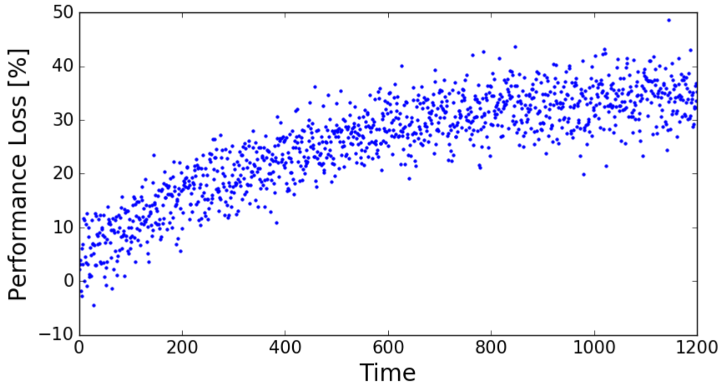


Figure 2.2: Example development of the performance loss over a dry-docking period given that no intermediate husbandry actions are taking place. Time is in days and noise is added.

The vessel hull and propeller performance can be estimated by comparing the actual measured power consumption with the theoretically determined value given the same conditions. In Carlton (2011), a comprehensive overview and discussion of vessel performance monitoring methods are presented. One crude method is to calculate the *Admirality Coefficient*, A_C as seen in Equation (2.7), where Δ is the displacement of the vessel, V is the speed and P_S is the total shaft power. This method does not account for environmental conditions and should only be used to compare loading conditions.

$$A_C = \frac{\Delta^{2/3} V^3}{P_S} \tag{2.7}$$

2.3 Vessel Data

All data about the vessel is extracted from three different datasets,

1. Measured vessel data, 338 parameters logged at different sampling frequencies from May 2014 to December 2016. Do notice that there are missing data for several periods in some or all sensors in this period.

2. Automatic identification system (AIS) data. Includes information of global positioning system (GPS) position of the vessel every couple of hours.

3. Computational fluid dynamics (CFD) information about the speed-power relation of the vessel in calm water.

In addition to this, information about the maintenance and retrofit activities are available. These activities can be seen in Table 2.2. Hopefully, we will be able to see the impact of these activities in the data.

Activity	**Date**
Repainting and propeller-polishing	2014-05-27
Propeller-polishing	2015-04-19
Propeller-polishing	2016-03-15

Table 2.2: Maintenance and retrofit activities

The data starts from right after the vessel was at dry-dock, being repainted. Due to this, the performance of the vessel is not expected to decrease significantly in the start.

2.3.1 Variable Selection

Depending on the topic to be investigated the variables that are of interest will vary a lot. Since this thesis is focused on the propulsion system of the vessel, a total of 25 variables has been selected from the 338 possible variables. This was mainly done to reduce the amount of data to a suitable amount such that analyzing the data would take shorter time and not exceed the computer capacity.

The selected variables with names, units, and number of observations can be found in Table 2.3. The selected variables are chosen based on experience, conversations with supervisors and DNV GL. There are of course other variables that could be of interest that are kept out, but to get a database suitable for analysis in the given time frame the amount of data had to be reduced.

Name	Unit	Observations
Cargo Level - Tank 1	m	16765
Cargo Level - Tank 2	m	20277
Cargo Level - Tank 3	m	20438
Cargo Level - Tank 4	m	24417
Sea Water Temperature	C	33577514
Atmospheric Temperature	C	25674429
Speed Over Ground	kts	7581
Speed Through Water	kts	7950
Wind Speed	kts	17982810
Wind Relative Direction	deg	2840350
Rudder Angle Port	deg	7062
Rudder Angle Starboard	deg	6629
Draft Forward	m	19809289
Draft Aft	m	16694584
Main Generator Engine 1 Power	kW	22809012
Main Generator Engine 2 Power	kW	19072520
Main Generator Engine 3 Power	kW	22044693
Main Generator Engine 4 Power	kW	17336655
Shaft Torque Port	kNm	47719517
Shaft Torque Starboard	kNm	47879705
Shaft Speed Port	rpm	13323090
Shaft Speed Starboard	rpm	12645440
Total Fuel Gas Flow to Main Generators	kg/h	1465695
Water Depth	m	2897521
Heading	deg	3089545

Table 2.3: Selected variables from the original dataset

From the variable selection seen in Table 2.3, we see that the number of observations for each variable range between a couple of thousand observations, to as much as 47 million observations. The rudder angles, speed over ground and speed through water are only logged roughly 7000 times over the full period. This means that on average they are only logged once every third hour. By inspecting the speed over ground and speed through water parameters we often see that they are logged with as much as 7-hour intervals.

Periods with missing data can be found by investigating how many kilobytes of data is stored each day for the full period, assuming a constant sampling frequency for all variables, and, in some extent, that all variables are logged. This can be seen in Figure 2.3.

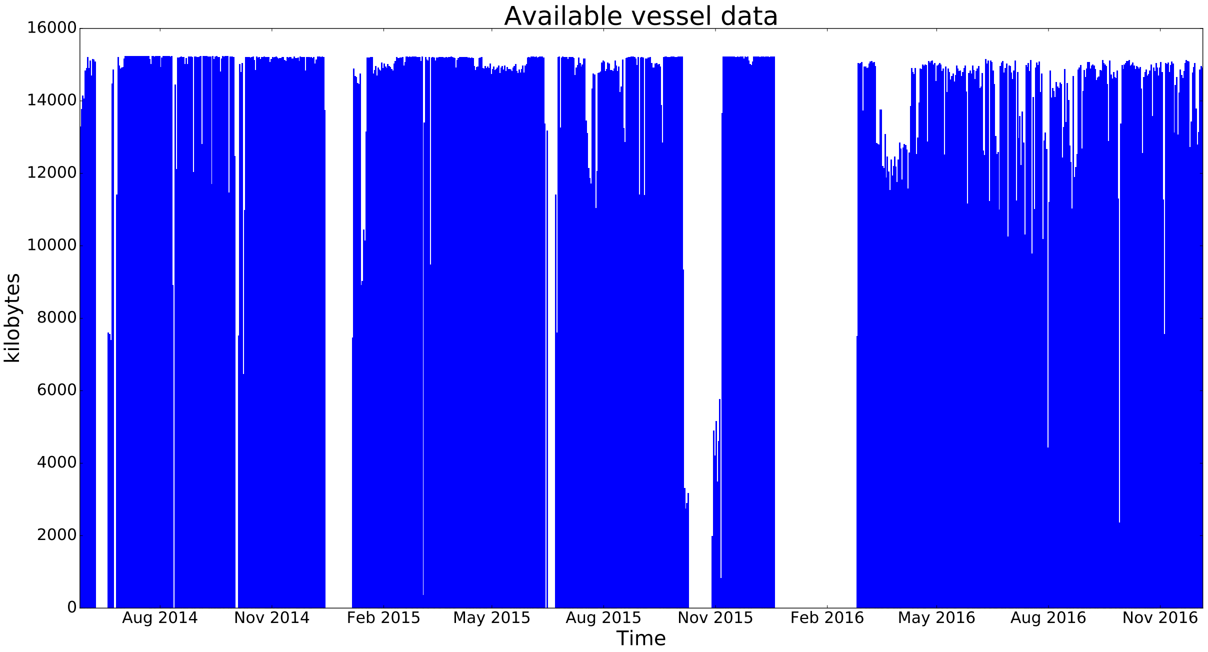


Figure 2.3: Kilobytes per day for the measured vessel data

In addition to this, there are periods where some variables behave strangely. One example of this is the wind speed seen in Figure 2.4. In the first year of data the wind speed is behaving as expected, but after May 2015 the wind speed is scaled down, then after November 2015, the wind speed is scaled up. Due to this, several periods of the data has to be completely disregarded for some or all of the variables.

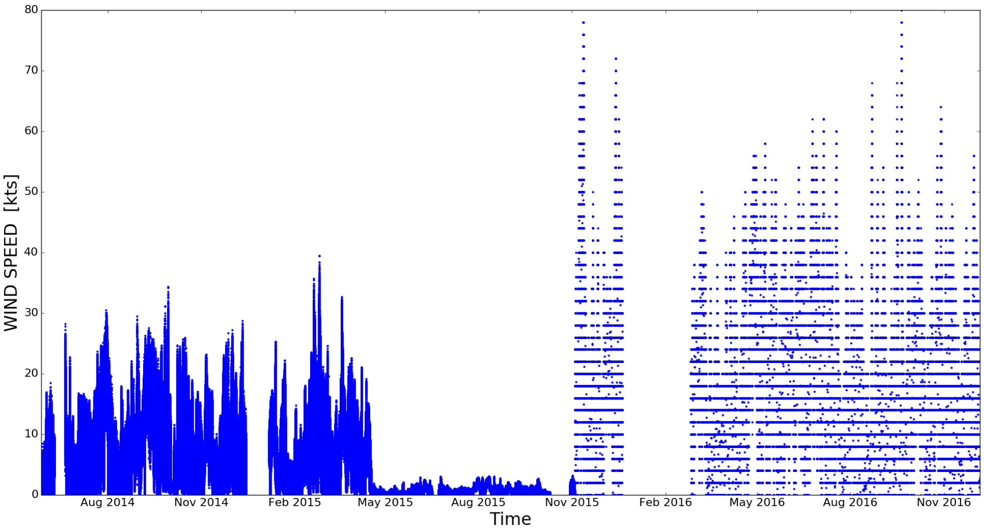


Figure 2.4: Measured wind speed

2.3.2 AIS Speed

In Table 2.3 we see that there are few speed over ground measurements from the original data. To obtain estimates of the speed it is possible to use the AIS data for the vessel. This data has information about the GPS position of the vessel as seen in Table 2.4. With this information, it is possible to calculate the estimated average speed of the vessel between two timestamps.

Variable	**Unit**
Latitude	Decimal degrees
Longitude	Decimal degrees

Table 2.4: Variables from the AIS data

To find the distance traveled between two GPS locations, we can use Vincenty's formulae described in Vincenty (1975). The formulae have been widely used in geodesy because they are accurate to within 0.5 mm on the Earth ellipsoid.

In Figure 2.5 the location of the vessel for the period we have data is shown. As we see, the vessel travels over long distances and possibly all sorts of weather. The GPS position of the vessel further makes it possible to use global reanalysis atmospheric data at a given time and position of the vessel.

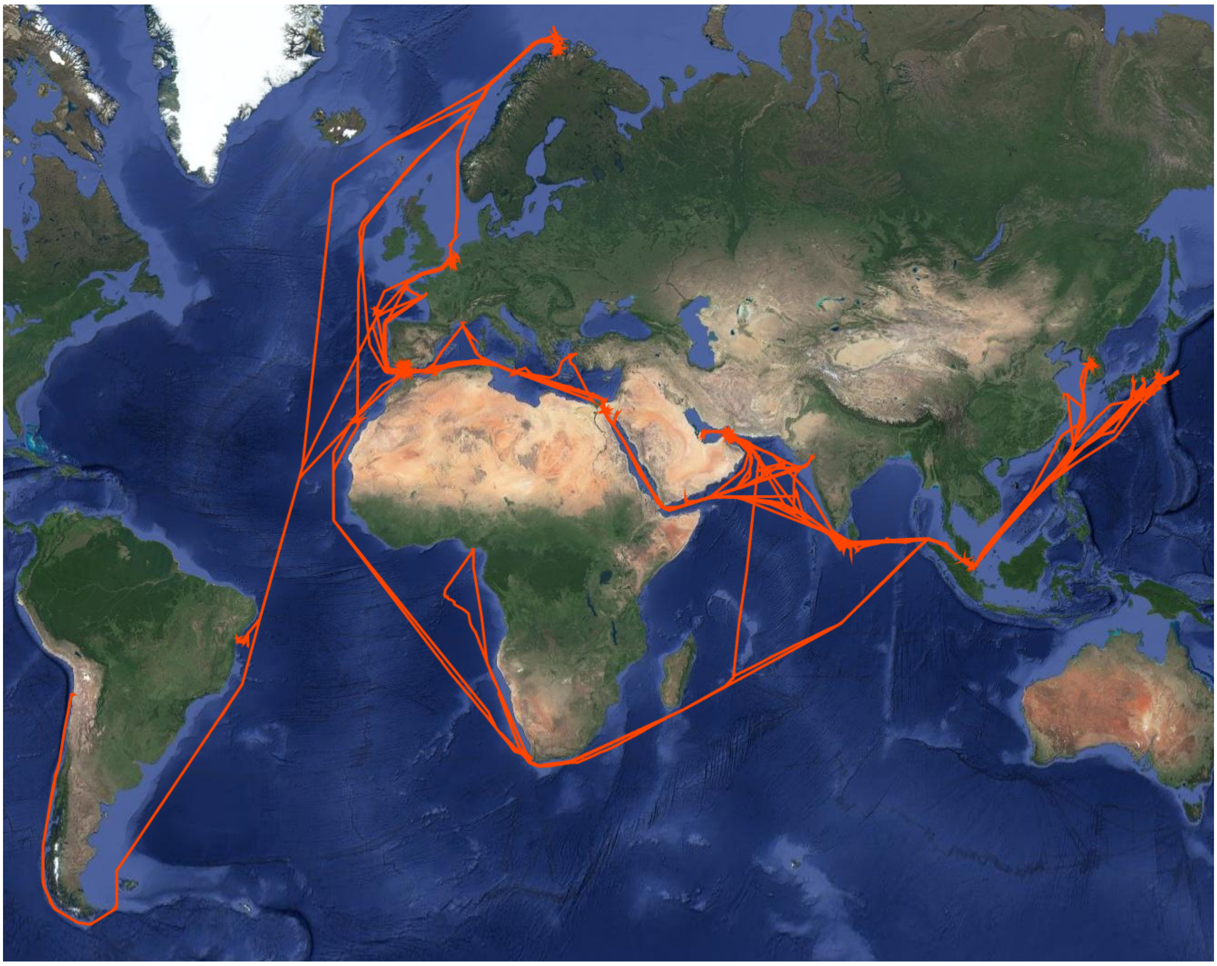


Figure 2.5: Vessel location for the available period

2.3.3 CFD Curves

Speed-power curves have also been calculated by DNV GL using computational fluid dynamics (CFD) for both laden (draft of 11.5 m) and ballast (draft of 9 m) conditions. These curves should be quite accurate for the case when the vessel is sailing in calm sea before any significant fouling has taken place (the only resistance on the vessel is the still-water resistance R_{SW}).

Based on these CFD curves we can make a polynomial curve-fit such that we obtain a formula which can give the expected speed for a given power consumption. If we then compare the measured speed to the expected speed as in Equation (2.6) we will have a PL to see the efficiency of the vessel. In Figure 2.6 the CFD speed-power curve is plotted with a second-degree polynomial curve fitted to the points. Both the fitted polynomials have an R-squared value above 0.995. The values from which these plots are made can be seen in Appendix B.1.

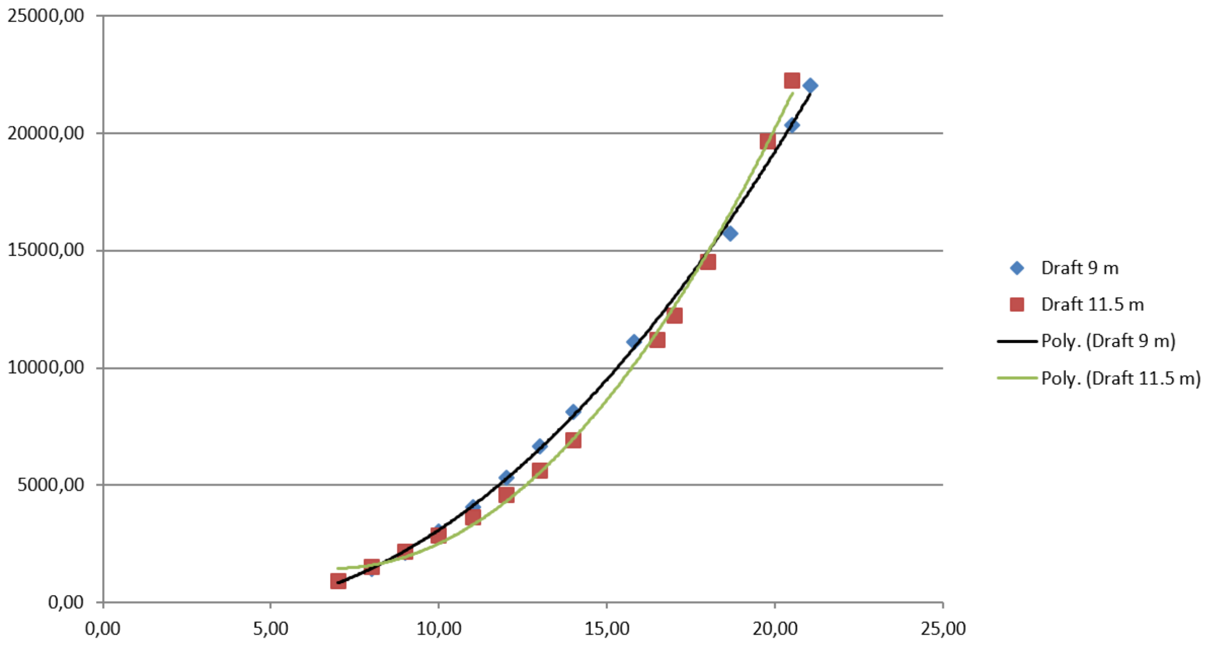


Figure 2.6: Speed-power curves for two different drafts calculated by CFD. X-axis represents the speed of the vessel [kts], Y-axis represents the shaft power consumption [kW].

2.4 Software Platforms

Handling large amounts of data and performing exploratory data analysis can be done by a variety of programming languages, Interactive Development Environments (IDEs) and software suites. Some popular languages include R, Python, C/C++, Java and Matlab and a commonly used software is Statistical Analysis System (SAS). In this work, I will utilize Python as the main platform for implementation and data analysis. This is due to several factors:

- Python is a free, interpreted, open-source platform with a large community (Python Foundation, 2017).
- Python can be augmented by a huge variety of free, open-source libraries and packages such as:
  - **Pandas**, an open-source Python library with powerful tools for handling and manipulating large amounts of data in an efficient manner (Pandas, 2017).
  - **SciPy**, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more (SciPy, 2017).

  - **SciKit-Learn**, an open-source Python library for machine learning and data mining (SciKit-Learn, 2017).
  - **Matplotlib**, an extensive plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms (Matplotlib, 2017).

When working with Python we will utilize Spyder, a free, open-source IDE for scientific programming in the Python language (Spyder, 2017).

There are of course several disadvantages with Python compared to other languages and tools. Since Python is a high-level interpreted language it is much slower than compiled languages like C and C++, but Python programs are in general shorter and more compact (Python Foundation, 1997). R was specifically developed for statistical use and has a richer set of libraries and packages for data science and more novel visualization possibilities than Python (DataCamp, 2017). However, R has a steep learning curve and gives little thought to memory management (Mwitondi, 2013), and was hence not chosen for this work.

Chapter 3

Statistical Learning Methods

In this chapter, several methods from the field of statistical learning are presented. The theory and interpretation are presented as well as simple examples for some of the methods. This chapter only presents a small subset of important methods from the broad field of statistical learning, and the selected methods based on practical use, interpretation and visualization. A brief introduction to statistical learning is given in Section 3.1. Section 3.2 covers preprocessing methods which are normally done on a dataset before any statistical analysis is carried out. In Section 3.3 unsupervised learning is presented. Learning in the supervised framework is discussed in Section 3.4. Section 3.5 presents methods and metrics for model assessment and selection.

3.1 Introduction

With statistical learning, we are talking about the ability to learn from data. Given a set of m observations, also called samples or objects,

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]^T, \tag{3.1}$$

where each observation $\mathbf{x}_i$ is a row vector of n variables, also called features or attributes,

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{in}], \tag{3.2}$$

Alternatively we could say we have a set of n variables,

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n], \tag{3.3}$$

where each variable $\mathbf{x}_i$ is a column vector of m observations,

$$\mathbf{x}_i = [x_{1i}, x_{2i}, \cdots, x_{mi}]^T \tag{3.4}$$

We want to be able to extract valuable information from the dataset $\mathbf{X} \in \mathbb{R}^{m \times n}$, represented as a matrix. As n gets larger than say 3, it becomes increasingly difficult to investigate and analyze the parameter interactions and underlying structure only by comparing one variable against another. Then it would be more beneficial to perform a multivariate analysis, which refers to statistical techniques used to analyze data that arises from more than one variable.

The general goal of the techniques would be to discover natural groupings in the data, variable correlations or to understand underlying dynamics. In the unsupervised framework described in Section 3.3, we discuss the case when only one set of data $\mathbf{X}$ and its internal structures are analyzed. Methods that are as transparent as possible have been chosen, allowing us to interpret and understand the results. Thus linear methods have been preferred over non-linear ones, avoiding black-box models. In the supervised framework described in Section 3.4, we discuss the case when the data is separated into two matrices $\mathbf{X}$ and $\mathbf{Y}$, and one wants to uncover the fundamental relationship between variables in $\mathbf{X}$ and variables in $\mathbf{Y}$. This can lead to models used for regression or classification.

In the literature, the unsupervised problem is less developed than the supervised one. With supervised learning there is a clear measure of success and you can more easily compare the effectiveness of different methods. With unsupervised learning there is no such direct measure of success (Hastie et al., 2001).

When performing exploratory data analysis on a large set of complex, real-world data, it is common to first explore the data using an unsupervised framework. This can reveal the structures and patterns of the data itself, often providing some important preliminary insight into the problem at hand. Then a supervised approach can be taken to establish and train models for regression and classification.

Many of the well-established data mining methods are designed to deal with data where the order of the observations has nothing to do with the pattern of interest. By adding the time dimension to the data, a new set of aspects and challenges arise as discussed by Last et al. (2004). This is kept in mind when applying the different methods since our observations are temporal sensory data.

3.2 Preprocessing

A central problem in exploratory data analysis on large datasets is the excessive amount of data. In many cases, there can be several hundred variables and millions of observations, or the opposite way around. Reducing the amount of data is often a necessary part of the preprocessing, to get a dataset that contains relevant information and that our computers can handle. For instance, in the Large Hadron Collider, there are more than 150 million sensors, delivering data 40 million times a second. 99.99995 % of this data is removed and they are left with 100 collisions per second of interest (CERN, 2009). As we have large amounts of raw data with high temporal resolution, we will need to reduce the amount of data while limiting the amount of information loss.

A crucial step in the analysis is to prepare the dataset for analysis. This usually involves mean-centering and scaling of the different variables, as well as proper outlier detection and handling. Various methods to reduce noise is also usually done. In some cases selecting samples and variables of interest is also important.

Outlier detection and handling are especially important since a percentage of the data will contain faulty values due to sensor failure, erroneous data storage and so on. By obtaining a clean dataset with a reduced amount of noise and anomalies one can greatly increase the effectiveness of the methods applied to the data.

3.2.1 Mean Centering

One of the most common preprocessing methods is mean centering. For each variable in the dataset, we want to center the column around zero by subtracting the mean of the column from each value. Mean centering can be defined for a data set $\mathbf{X} \in \mathbb{R}^{mxn}$

$$\bar{\mathbf{x}} = \frac{1}{m}\mathbf{X}^T\mathbf{1} \tag{3.5}$$

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^T, \tag{3.6}$$

where $\mathbf{X}_c$ is the mean centered dataset, $\mathbf{1} \in \mathbb{R}^m$ is a vector of ones and $\bar{\mathbf{x}} \in \mathbb{R}^n$ is a vector containing the mean value for each variable.

3.2.2 Scaling

Scaling methods are used to standardize the range of the variables in the dataset. It is also known as data normalization. This is useful when the variables are measured in different units or have different magnitudes. This allows each variable an equal opportunity to influence the result.

One common method is to scale each variable to have zero-mean and unit-variance. For each variable $\mathbf{c}_i$ we subtract its mean and divide by its standard deviation,

$$\tilde{\mathbf{c}}_i = \frac{\mathbf{c}_i - \bar{\mathbf{c}}_i}{\sigma_i}, \quad i = 1, 2, \ldots, n, \tag{3.7}$$

to obtain the scaled variable vector $\tilde{\mathbf{c}}_i$ with zero-mean and unit-variance.

3.2.3 Missing Values

For various reasons, many datasets generated from sensors have missing values. These missing values are often stored as Not-a-Number (NaN), zeros or blanks. Data sets with many missing values are incompatible with several data mining methods, like Principle Component Analysis using Single Value Decomposition (Martens and Martens, 2001). There are three methods to handle missing values:

1. Remove entire observations or variables containing missing values
2. Impute the missing values, i.e., to infer them from the known parts of the data
3. Use algorithms that can work with missing data

The second approach is preferable when there are few, unstructured missing values without too much variation in the data set. When there are many structured missing values simply inserting the mean, median or interpolating between known values might give unwanted results. There are of course more sophisticated methods available. In *Statistical Analysis* *with Missing Data*, Little and Rubin (2014) discusses several methods for imputing missing values, for instance, a maximum likelihood estimation. Using algorithms that can work with missing data might be a good choice in some cases. For instance, Principle Component Analysis can be run on a data set with < 5 % missing values of the data, by using a modified version of the NIPALS algorithms (Martens and Martens, 2001).

When you are logging time series from various sensors the sampling frequency of the sensors can often vary. If we were to put several sensors in a matrix form, with rows as instances of time, the matrix might become sparse, meaning there are might only be a few sensors that logged at the exact same timestamp. These are not missing values but simply an effect of having different sampling frequencies for the sensors. To avoid the problem of a sparse matrix there are different approaches. One can divide the data set into groups based on the frequency of the sensors, or simply round the timestamps to a specific nearest value, thus trying to align different measurements on one timestamp. An implementation of this is done in Section 4.4

3.2.4 Noise

Noise is variance or random error occurring in the data. Noise is often assumed to have a Gaussian distribution. It can often be useful to remove noise by smoothing the data signal. For smoothing the data there are many methods that can be used. I will explain one of them in this section.

Moving average is a method to reduce both the size and the noise of a vector. The method can be viewed as a low-pass filter essentially removing high-frequency noise. Moving average creates series of averages of different subsets of the full dataset. One way to do this is the centered moving average (CMA). In CMA one uses a window of length k, which is centered around each data point. Each data point is then replaced by the unweighted average of the points within its respective window. Suppose we have a vector $\mathbf{x}$ with m values. The centered moving average is then given by,

$$\text{CMA} = \frac{1}{2m+1} \sum_{i=-m}^{m} \mathbf{x}_i \tag{3.8}$$

where $n = (k-1)/2$. In the start and end of the vector, the window around the data points is not fully defined. Due to this the CMA is either not computed or computed by only using the available values within the window. In Figure 3.1 an example of centered moving average is shown on a noisy sine-function with four different window sizes and 200 data points. We clearly see the influence of the choice for k. If we choose k too small the noise does not get filtered enough, choose k to large and we loose information about the signal.

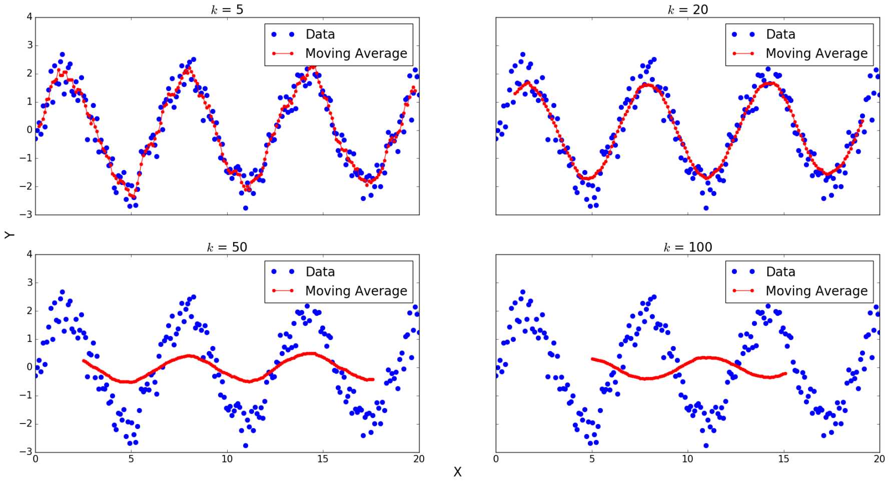


Figure 3.1: Moving average with four different window sizes.

There are various methods one can use to calculate the moving average, for instance, weighted moving average or exponential moving average as explained in Wikipedia (2017). By using a central window instead of a backward looking window, we do not introduce a phase lag in the time series. Other methods to reduce noise are for instance the Kalman filter described in Kalman (1960). The Kalman filter does not make any assumption that the errors are Gaussian. However, the filter yields the exact conditional probability estimate in the special case that all errors are Gaussian-distributed.

3.2.5 Outliers

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism (Breunig et al., 2000). Outlier detection, also known as anomaly detection is the detection of outliers in the dataset. This must not be confused with novelty detection where we have a clean dataset and wish to detect anomalies in new observations. Outliers are often categorized into three different types (Han et al., 2011):

1. Global outliers, where an observation deviates significantly from the rest of the data
2. Contextual outliers, where an observation deviates significantly based on a given context
3. Local outliers, where an observation deviates significantly from the other observations in the neighborhood

Both detection and handling of outliers is a task that has no universal method that works perfectly in all situations. Outlier detection algorithms can be grouped into two categories:

1. Model-based algorithms
2. Data-based algorithms

Model-based methods often assume that observed data are governed by some statistical distribution (e.g., Gaussian, Poisson *etc.*) with appropriate parameters describing the distribution. We then identify outliers based on how unlikely the given data is, based on the chosen distribution. Data based techniques, on the other hand, attempt to avoid model assumptions; relying on the concepts of distance, density of points or other concepts like the angle based approach described by Kriegel et al. (2008).

We will not go into detail on outlier methods, but one simple model based algorithm is the box plot described by Tukey (1977). This is a simple way of finding and removing outliers from the dataset. A simple data-based method to filter outliers in a high-dimensional dataset is the k-Nearest Neighbor algorithm described by Ramaswamy et al. (2000). The method ranks each observation on the basis of the distance to its k nearest neighbors (kNN) and declares the top n points in this ranking to be outliers, where n has to be specified. One of the most commonly used outlier detection methods is simple min-max filters, where all values outside a given range are considered outliers.

3.3 Unsupervised Learning

The unsupervised learning problem is the machine learning task concerned with revealing internal structures in unlabeled data. In the unsupervised framework, we observe only the data itself with no relation to a measured outcome. Our task is rather to describe how the data is organized.

3.3.1 Latent Variable Analysis

Latent variable methods are used to reduce the dimensionality of a dataset $\mathbf{X}$ with minimal loss of information. The method identifies the directions of maximum variance in a high-dimensional data set and projects it onto a smaller dimensional subspace while retaining most of the information. This can make the data easier to explore and visualize.

We can illustrate this with an example. Assume we have a dataset $\mathbf{X} \in \mathbb{R}^{m \times n}$, containing information about m different cars which we call observations. Let us say we have identified and measured n different properties like displacement, power, weight and so on. These properties are called variables. Many of these variables will be correlated and thus redundant in the context of reducing the dimension of the data set. With latent variable analysis, we want to reveal latent structures and summarize each car with fewer variables. This is achieved by constructing new variables using linear combinations, often called latent variables (LVs), of the old ones, for example, displacement minus power. These variables then span a subspace, called the LV-space, of the original variable space which has reduced dimensionality.

A bilinear subspace model $\hat{\mathbf{X}}$ of the data set $\mathbf{X} \in \mathbb{R}^{m \times n}$ can be expressed as

$$\hat{\mathbf{X}} = \mathbf{Z}\mathbf{P}^T. \tag{3.9}$$

The model is expressed in terms of a score matrix $\mathbf{Z} \in \mathbb{R}^{m \times a}$ and a loading matrix $\mathbf{P} \in \mathbb{R}^{n \times a}$. Where a is the model complexity, or in other words the dimension of the model subspace. The error of the model is contained in the residual matrix $\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}$.

The rows of the score matrix $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_a]$ are called scores and contain coordinates of the projections of the observations onto the loading vectors. The scores can be used to visualize relationships among observations. (In the car example, the score could show that one car has a larger LV, displacement minus power, than the other.)

The rows of the loading matrix $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_a]$ are called loadings and contain the axis of a latent variable in the X-space. The set of loadings constitute a basis for the LV-space. The loadings are used to visualize relationships between the variables. (In the car example the loadings could show that power and displacement are correlated and the direction in which the LV, displacement minus power, is pointing in the X-space.)

3.3.2 Principal Components

Principal Component Analysis (PCA) is a common unsupervised latent variable analysis used to reveal internal structures in the dataset. The method goes as far back as 1901, invented by Karl Pearson (Pearson, 1901), but was independently developed and named by Harold Hotelling in the 1930s (Hotelling, 1933).

PCA assumes the dataset can be represented as a linear combination of the variables, which of course will not always be the case. This suggests the results will be best if observations are selected with similar conditions such that the PCA model is linearized around this operating condition. Alternatively, several areas of research have explored how applying a nonlinearity prior to performing PCA could extend this algorithm, this has been termed kernel-PCA (Schölkopf et al., 1997). These nonlinear methods are often difficult to interpret and understand, and will not be explored in this thesis.

Another assumption is that the mean and variance are sufficient statistics, i.e that they entirely describe a probability distribution. The only zero-mean probability distribution being described by its variance is the Gaussian distribution. In order for this assumption to hold, the probability distribution of the variables must be Gaussian (Shlens, 2014).

PCA also assumes that large variances have important dynamics, that components with larger variance correspond to interesting dynamics and lower ones to noise. Several preprocessing steps should be done prior to the analysis. Mean centering is important since PCA is based on the covariance matrix which is formed from centered data. The analysis is also sensitive to the relative scaling of the variables since it is based on the least squares method. Scaling is useful if we want the variables to have an equal influence on the result, regardless of magnitude and unit. If we have prior knowledge about the variables and how much influence we would like them to have, weighting can also be considered.

The PCA procedure consists of finding the a principal components in decreasing order of explained variance of $\mathbf{X}$, under the constraint that each component is orthogonal to the preceding components. The maximum number of principal components that can be extracted from the data is n, the number of variables. The bilinear PCA model can be written (Martens and Naes, 1992) as

$$\mathbf{X} = \mathbf{Z}\mathbf{P}^T + \mathbf{E} = \sum_{i=0}^{a} \mathbf{z}_i \mathbf{p}_i^T + \mathbf{E}, \tag{3.10}$$

where the loading matrix **P** transforms the data set **X** to the uncorrelated orthogonal basis set **Z**, or scores matrix

$$\mathbf{Z} = \mathbf{X}\mathbf{P}. \tag{3.11}$$

We now want to project all observations of **X** onto the ith loading vector $\mathbf{p}_i$ to get the ith score vector $\mathbf{z}_i$ in a way that maximizes the variance of the projections. To achieve this we maximize

$$\begin{aligned} \text{maximize} \quad & \text{Var}[\mathbf{z}] = \text{Var}[\mathbf{X}\mathbf{p}] = \mathbf{p}^T\mathbf{X}^T\mathbf{X}\mathbf{p} = \mathbf{p}^T\Sigma\mathbf{p} \\ \text{subject to} \quad & \mathbf{p}^T\mathbf{p} = 1. \end{aligned} \tag{3.12}$$

By use of Lagrange multipliers it can be shown that the maximization problem reduces to an eigenvalue problem (Höskuldsson, 1994). Where the ith loading vector $\mathbf{p}_i$ is the eigenvector with the ith largest eigenvalue of the covariance matrix Σ. Thus we can find the projections $\mathbf{z}_i$ of **X** onto $\mathbf{p}_i$, which is called the scores.

Example

To illustrate PCA we choose to work with a dataset consisting of 10 cars, or observations with 4 variables each. The data is retrieved from US Environmental Protection Agency (2017) and consists of data on car models for 2017. The selected variables for each car are horsepower, engine displacement, weight, and CO_2 emission. By purpose, five high-end sports cars and five regular cars have been chosen. The dataset can be seen in Table 3.1. Since PCA is unsupervised it is not aware of this difference, but hopefully, the results will show the difference between two classes of cars. To cope with different units for the variables the data is mean centered and scaled to have zero mean and unit variance, as described in Sections 3.2.1 and 3.2.2.

Car model	**Power [hp]**	**Engine disp [l]**	**Weight [lbs]**	**CO$_2$ (g/mi)**
Audi A4	211	1.984	3875	314
Hyundai Elantra	128	1.4	3125	250.38
Subaru Galaxy	175	2.5	3875	264.27
Toyota Corolla	132	1.798	3125	246.19
Volkswagen Golf	170	1.798	3375	174.55
Aston Martin Vanquish	568	6	4500	309.22
BMW M5	553	4.4	4750	602.28
Chevrolet Corvette	650	6.2	3875	280.2
Mercedes S65	621	5.98	5500	646.63
Porsche 911 Turbo S	572	3.8	3875	264.87

Table 3.1: Data on 2017 car models. Retrieved from US Environmental Protection Agency (2017).

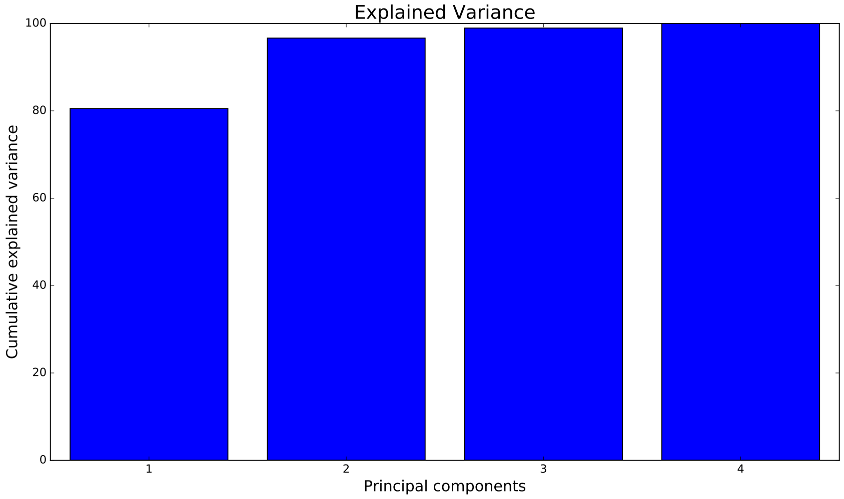


Figure 3.2: Cumulative explained variance for principal components

In Figure 3.2 we can see how much of the variance is explained for each principal component added to the model. We see that the model is able to explain 96.6 % of the variance in the data using only the first two principal components, which is a strong result. Hence, we plot the scores and loadings for the for the two first components.

The scores plot seen in Figure 3.3 shows the projection of the cars onto the LV-space. The cars are colored red and blue for sports cars and regular cars respectively. We can clearly see a natural grouping of similar cars. The leftmost cluster contains all the regular cars, the middle cluster contains light sports cars and the right cluster contains the heavy sports cars. We also note that the normal cars appear in a cluster with higher density than the other, suggesting that they are more similar to each other than the sports cars.

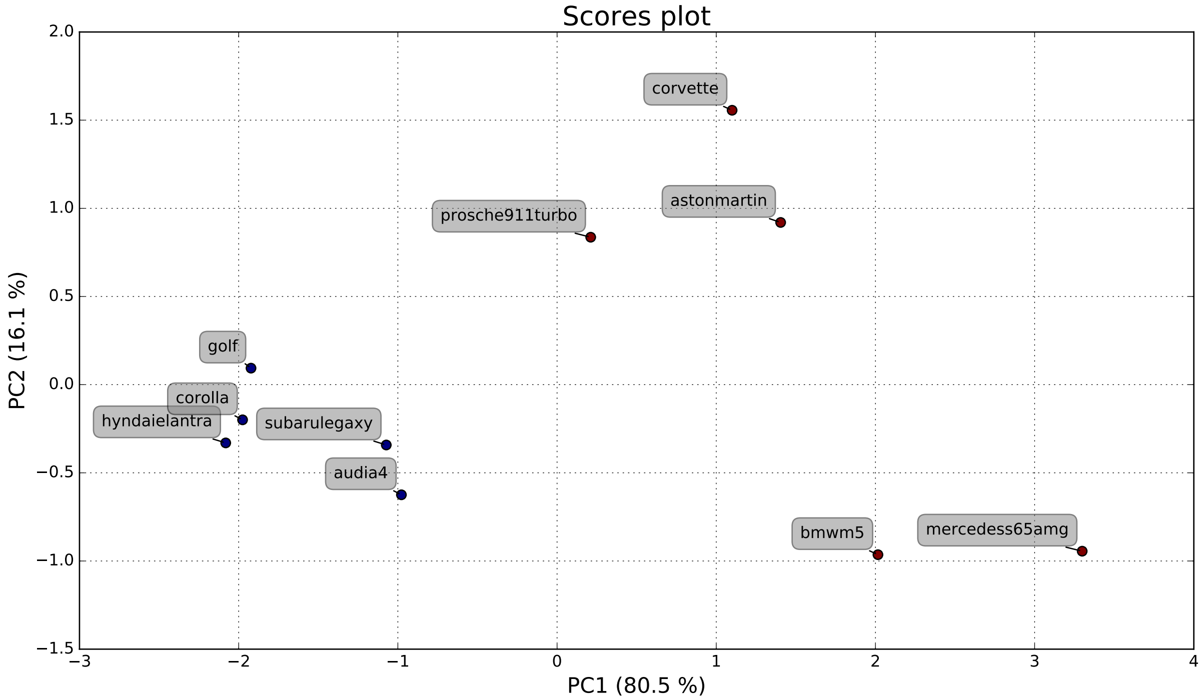


Figure 3.3: Scores plot for PC1 and PC2

The loadings plot in Figure 3.4 show the variables spanning the LV-space. Note that the PC1-axis starts from 0.45, thus all variables give a positive contribution for PC1. This means all variables are positive correlated to some degree, where displacement and horsepower show the strongest correlation. In other words, the tendency is that the larger a variable is the larger the other variables are. A larger engine usually means more weight and more emissions.

PC2 shows that the engine displacement and horsepower are negatively correlated with emission and weight, however, note that this PC only accounts for 16 % of the explained variance. More powerful cars with low emissions and less weight will have a larger PC2 than others.

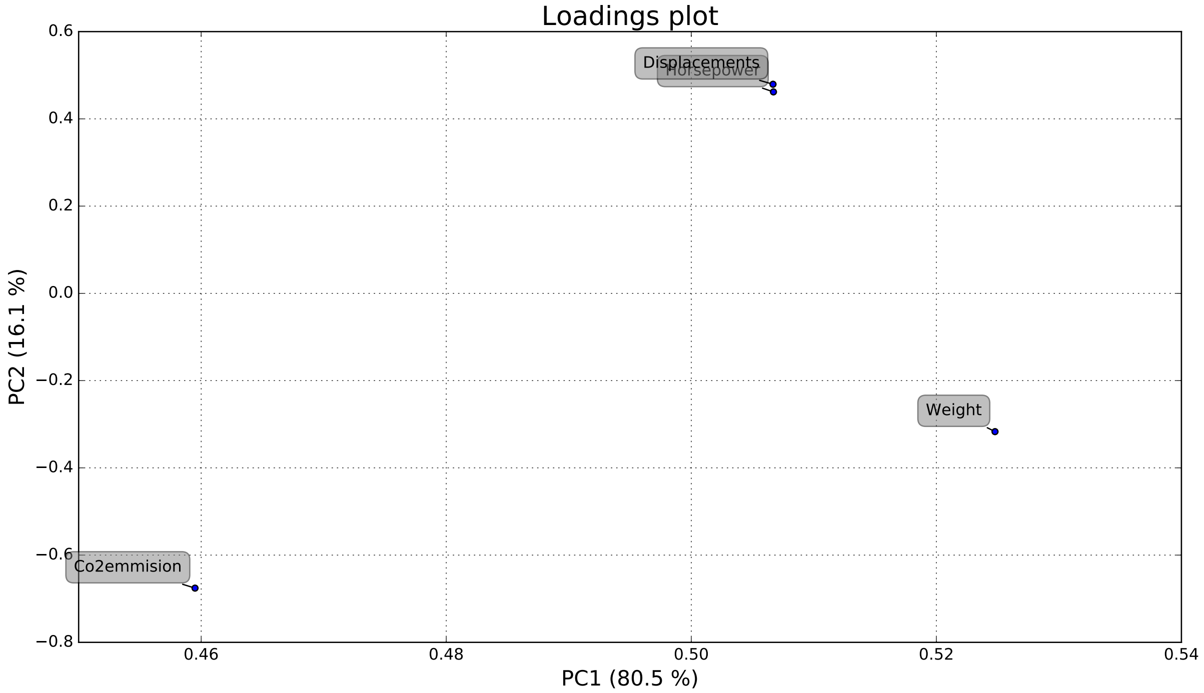


Figure 3.4: Loadings plot for PC1 and PC2

3.4 Supervised Learning

In supervised learning we are concerned with finding a model that best describes the relationship between some data **X** and **Y**. To construct such a model one has to use a set of training data $\mathscr{T}$, where each sample is said to be a pair consisting of an input observation $\mathbf{x}_i$ and a desired output $\mathbf{y}_i$. This is often called labeled data. The model is then trained to fit the data and can be used to perform classification or prediction on new unlabeled samples that were not used to train the model. In this thesis, we will explore regression models, models that predict quantitative outcome labels $\mathbf{y}_i$.

3.4.1 Linear Regression Methods

Linear models were largely developed in the precomputer age of statistics, but even in today's computer era, there are still good reasons to study and use them. They are simple and often provide an adequate and interpretable description of how the inputs affect the output. For prediction purposes, they can sometimes outperform fancier non-linear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data (Hastie et al., 2001). Linear regression models assume that the regression function $\mathrm{E}(\mathbf{y}|\mathbf{X})$

is linear in the inputs, $\mathbf{x}_1, \ldots, \mathbf{x}_n$. In addition, various transformations and basis-expansions of the input can be applied to expand their scope while still retaining the linearity. Such transformations can for instance be $\mathbf{x}^2$, $\sqrt{\mathbf{x}}$, and $\log(\mathbf{x})$. The mathematical descriptions and derivations of the linear models for regression presented here are based on Chapter 3 in *The Elements of Statistical Learning* by Hastie et al. (2001).

3.4.1.1 Ordinary Least Squares Regression

Suppose the input vector $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ of n variables. We want to predict the real-valued output $y \in \mathbb{R}$. The linear regression model can be written as

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{n} \mathbf{x}_j \beta_j, \tag{3.13}$$

The linear model assumes that the regression function $\mathrm{E}(\mathbf{y}|\mathbf{X})$ is linear, or that the linear model is a reasonable approximation. The β_j's are the unknown parameters of coefficients to be determined, and β_0 will be the intercept of the model. The input variables $\mathbf{x}_j$ can come from different sources:

- quantitative inputs
- transformations of quantitative inputs, such as $log(\mathbf{x}_1)$ or $\sqrt{\mathbf{x}_2}$
- basis-expansions of quantitative inputs, such as $\mathbf{x}_2 = \mathbf{x}_1^2$, $\mathbf{x}_3 = \mathbf{x}_1^3$
- interactions between quantitative inputs, such as $\mathbf{x}_3 = \mathbf{x}_1 \cdot \mathbf{x}_2$

No matter the source of $\mathbf{x}$, the model is still linear in its parameters β_j.

With a set of training data $(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_m, y_m)$ we can estimate the coefficients β_j using the least squares method. That is, choose the coefficients β_j such that the residual sum of squares (RSS),

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{m} (y_i - f(\mathbf{x}_i))^2 \\ &= \sum_{i=1}^{m} (y_i - \beta_0 - \sum_{j=1}^{n} x_{ij} \beta_j)^2, \end{aligned} \tag{3.14}$$

is minimized. Equation (3.14) makes no assumptions about the validity of model (3.13) but simply finds the best linear fit to the data. Least squares fitting is intuitively satisfying no matter how the data arise; the criterion measures the average lack of fit. Figure 3.5 illustrates the geometry of least-squares fitting in the $\mathbb{R}^{n+1}$-dimensional space occupied by the pairs (**X**, **y**).

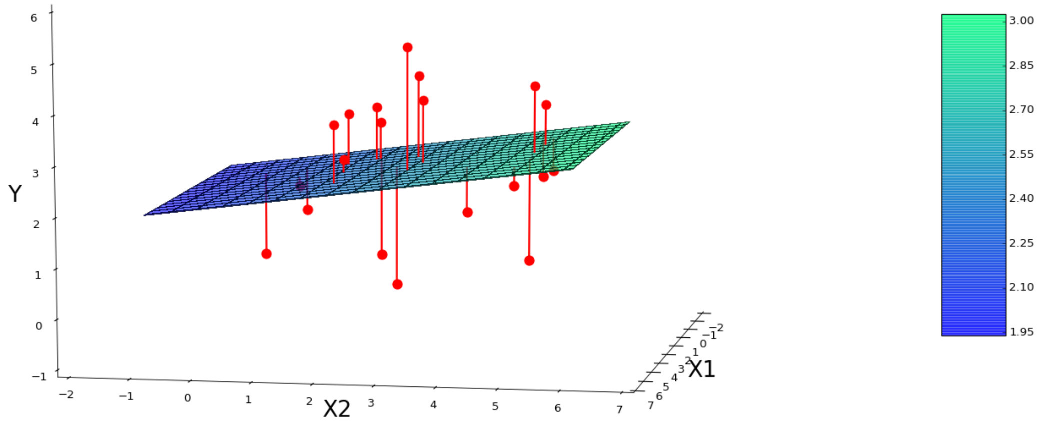


Figure 3.5: Linear least squares fitting with $\mathbf{X} \in \mathbb{R}^2$. We seek the linear function of **X** that minimizes the sum of squared residuals from **y**

If we let $\mathbf{X} \in \mathbb{R}^{m\times(n+1)}$ be a matrix where each row is an input vector with a 1 in first position, and similarly let $\mathbf{y} \in \mathbb{R}^m$ be the vector of outputs in the training set, then the RSS can be re-written as

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta). \tag{3.15}$$

This function is quadratic in its $n+1$ parameters. If we assume that **X** has full column rank, we can differentiate (3.15) and set it equal to zero

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0 \tag{3.16}$$

to obtain the unique solution

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.17}$$

Now the predicted values of the training inputs are given as

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3.18}$$

where $\hat{\mathbf{x}}_i = \hat{f}(\mathbf{x}_i)$. If the columns of $\mathbf{X}$ are not linearly independent ($\mathbf{X}$ does not have full columns rank) then $\mathbf{X}^T\mathbf{X}$ is singular and the least squares coefficients $\hat{\beta}$ are not uniquely defined.

Lets now assume that the observations $\mathbf{x}_i$ are not random and that the output values $\mathbf{x}_i$ have a constant variance σ^2 and are uncorrelated. The variance-covariance matrix of the least squares coefficients estimates $\hat{\beta}$ can now be derived from (3.17) as

$$\mathrm{Var}(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \tag{3.19}$$

where a typical estimate of the variance, σ^2, is given by

$$\hat{\sigma}^2 = \frac{1}{m-n-1}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2. \tag{3.20}$$

The $m-n-1$ in the denominator makes $\hat{\sigma}^2$ an unbiased esitmate of σ^2, $E(\hat{\sigma}^2) = \sigma^2$. If we further assume that (3.13) is the correct model of the mean and that the deviations of Y around its expected values are additive and Gaussian, we can write

$$Y = \beta_0 + \sum_{j=1}^{n}\mathbf{x}_j\beta_j + \epsilon, \tag{3.21}$$

where $\epsilon \sim N(0,\sigma^2)$. Hence, we can see from (3.21) that $\hat{\beta} \sim N(\beta,(\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$. This is a multivariate normal distribution or multivariate Gaussian distribution which is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions. Under these assumptions we can use the distributional properties to form confidence intervals for β_j and hypothesis tests. To test the hypothesis that a particular coefficient $\beta_j = 0$, we form the Z-score

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}, \tag{3.22}$$

where v_j is the jth diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$. Under the null hypothesis, that is $\beta_j = 0$, z_j is distributed as a t distribution with $m-n-1$ degrees of freedom. From this we see that a large absolute value of z_j will lead to a rejection of this null hypothesis. If we have large enough samples, say $m > 100$ the tail quantiles of a normal distribution and a t-distribution becomes negligible, and so we normally use the normal quantiles. By forming a null hypothesis

$$H_0 : \beta_j = 0$$
$$H_1 : \beta_j \neq 0$$

we can use z_j to reject H_0 if the corresponding probability-value is less than a given threshold α (typically 0.05). In other words, if the probability of attaining a particular β_j is very low under the assumption of H_0, say 5 %, we reject H_0 and assume H_1, and we can say that the result is significant. Note that this will also give a false rejection of the null hypothesis in 5 % of the cases.

3.4.2 Nonlinear Methods for Regression

In Section 3.4.1 we assumed that the regression function $E(\mathbf{y}|\mathbf{X})$ is linear in the inputs, $\mathbf{x}_1, \cdots, \mathbf{x}_n$. In regression problems the true function $f(\mathbf{X}) = E(\mathbf{y}|\mathbf{x})$ is often not linear in its inputs $\mathbf{X}$. The true function $f(\mathbf{X})$ will often be nonlinear and nonadditive in $\mathbf{X}$. By the use of nonlinear methods we may get better predictions, but at the cost of interpretability (Hastie et al., 2001). Many of the nonlinear methods, such as neural networks, have several tuning parameters, and it can be difficult to get the correct tuning. In this section we will focus on the nearest neighbor regression, an unstructured method that makes no assumptions on the model of the true function $f(\mathbf{X})$.

3.4.2.1 Nearest Neighbors Regression

Nearest-neighbor methods use those observations in the training set $\mathcal{T}$ closest in input space to x to form $\hat{Y}$. Specifically, the k-nearest neighbor fit for $\hat{Y}$ is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{3.23}$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample. x and y can be both scalars and vectors in this equation. Closeness implies a metric, which for instance can be Euclidean distance. Other metrics for closeness and calculation of the distance matrix can be found in Appendix A. So, in words, we find the k observations with x_i closest to x in input space, and average their responses.

The k-nearest neighbor estimate of y is the average response of the k closest observations x_i to x. If the observations x are no uniformly distributed, it is normal to weight the x_i based on their distance to x as seen in Equation (3.24)

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \omega_i, \quad \omega_i \propto \frac{1}{d_i} \tag{3.24}$$

where d_i is the distance from x to x_i.

The choice of k determines the complexity of the model. If $k = 1$ each point will only consider itself as is its own closest neighbor. Hence the error in the training data set will always be 0 for $k = 1$. The model appears to have only one parameter k. However the effective number of parameters can be estimated as m/k, and in general, this number is larger than the n parameters we have in least-squares models as described in Section 3.4.1.1. This also means that the model complexity decreases when the number of neighbors k increases. Even if our model (3.23) is simple to understand, and only has one tunable parameter k, we do not get a good understanding of the nature of the relationship between $\mathbf{X}$ and $\mathbf{y}$. Nearest Neighbor regression has proven good in low-dimensional features but should be avoided for some high-dimensional features due to the bias-variance tradeoff (Hastie et al., 2001) which we will discuss further in Section 3.5.1. To find an optimal k we can utilize cross-validation. Here we pick a k that minimizes the test error, which is a reliable and effective way of selecting k. Cross-validation will be further discussed in Section 3.5.2.

3.5 Model Assessment and Selection

Model selection refers to estimating the performance of different models in order to choose the best one. Having chosen a final model, model assessment refers to estimating its prediction error (generalization error) on new data. Model assessment and selection usually involves finding the optimal model complexity (i.e how many features to include), as well as estimating the model stability and predictive ability for a statistical model.

3.5.1 Bias-Variance Tradeoff

Consider a target variable $\mathbf{y}$, an input vector $\mathbf{X}$ and a prediction model $\hat{f}(\mathbf{X})$ which has been estimated from the training set $\mathscr{T}$. Let $L(\mathbf{y}, \hat{f}(\mathbf{X}))$ measure the loss between $\mathbf{y}$ and $\hat{f}(\mathbf{X})$. Typical choices for the loss are

$$L(\mathbf{y}, \hat{f}(\mathbf{X})) = \begin{cases} (\mathbf{y} - \hat{f}(\mathbf{X}))^2 & \text{squared error} \\ |\mathbf{y} - \hat{f}(\mathbf{X})| & \text{absolute error.} \end{cases} \tag{3.25}$$

The performance of a model is often assessed by the test error. The test error is the prediction error when used on an independent test set where $\mathbf{X}$ and $\mathbf{y}$ are drawn randomly from their joint distribution. The test error is conditional on the fixed training set $\mathscr{T}$ that were used to estimate the model and can be expressed as

$$\text{Err}_{\mathscr{T}} = \text{E}[L(\mathbf{y}, \hat{f}(\mathbf{X}))|\mathscr{T}]. \tag{3.26}$$

The test error is often called generalization error, as it describes how well the model generalizes to new data. In most cases, we estimate the expected test error

$$E[\text{Err}_{\mathscr{T}}] = \text{E}[L(\mathbf{y}, \hat{f}(\mathbf{X}))] = \text{Err} \tag{3.27}$$

Note that this expression averages over everything that is random, including the randomness in the training set that predicts $\hat{f}$. Additionally, the training error is the average loss over the training set

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{m} L(y_i, \hat{f}(\mathbf{x}_i)). \tag{3.28}$$

We now define error due to bias and error due to variance as done by Fortmann-Roe (2012):

- **Error due to bias** is the difference between the expected value of our prediction and the true target value. Bias in general measures how far off our prediction is from the correct value. A high bias can cause a model to miss relevant relationships between inputs and target outputs, which is called underfitting.

- **Error due to variance** is the expected squared deviation of a prediction around its mean, i.e the variability of a model prediction. A high variance can lead to the modelling of random fluctuations in the training data, which is called overfitting.

We normally see that, as the complexity of a model $\hat{f}$ increases, the bias decreases while the variance increases, and vice versa when the complexity decreases. If we assume that $\mathbf{y} = f(\mathbf{X}) + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon^2)$, we can derive an expression for the expected prediction error of a regression fit $\hat{f}(\mathbf{X})$ at an input $\mathbf{X} = x_0$, using the squared error loss:

$$\begin{aligned}\mathrm{Err}(x_0) =& E[(\mathbf{y} - \hat{f}(x_0))^2 | \mathbf{X} = x_0] \\ =& \sigma_\epsilon^2 + [E[\hat{f}(x_0)] - f(x_0)]^2 + E[\hat{f}(x_0) - E[\hat{f}(x_0)]]^2 \\ =& \sigma_\epsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\ =& \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}.\end{aligned} \tag{3.29}$$

The first term is the variance of the target around its true mean and is unavoidable. The second and third terms are due to the previously explained bias and variance. We then select a model $\hat{f}_\alpha$ that minimizes the expected test error Err. The α denotes a tuning parameter for the model (for example, $\alpha = k$ in nearest-neighbor regression).

3.5.2 Cross-Validation

Cross-Validation (CV) is a method to determine different characteristics of a model. It can be used to find a suitable number of components for a model, as well as the reliability and predictive ability of the model (Martens and Martens, 2001). It is important to have samples in the training data i.e $\mathbf{X}$ and $\mathbf{y}$ that contain information explaining all the types of variation to be modeled, otherwise a seemingly good model for the training data will struggle to give sound predictions for new samples. This is what CV intends to reveal.

Typically we split the data into three parts: a training set, a validation set, and a test set. The training set is used to fit the model, the validation set is used to estimate the prediction error for model selection and finally, the test set is used to assess the test error over an independent test set for the final chosen model (Hastie et al., 2001). This way of splitting the data can be seen at the top of Figure 3.6. Cross-validation uses a slightly different approach in that it only splits the data into two parts at fist, a training, and a test set. The cross-validation method is widely used to estimate expected prediction error Err. Specifically, we will consider the K-fold cross-validation.

The approach is simple, instead of setting aside an independent validation set to estimate the prediction error, we split the data pseudo-randomly into K equally sized parts. Each of the K parts is then used as a validation set once. In other words, for each part k we use the remaining $K-1$ parts to create our model and then use k to validate the model. This process is then repeated K times such that all data is used for both training and validation. This gives us K estimates of the prediction error and K estimates of the training error. This will again give us the possibility to calculate the mean and standard deviation of the estimated prediction errors. The way cross-validation splits the data into segments is visualized in Figure 3.6.

Stable estimates through iterative processes suggest the model is reliable, whereas well predicted hidden samples suggest the model has a good predictive ability. However, it is important to note that the reliability and predictive ability only holds for samples similar to the ones in the training set.

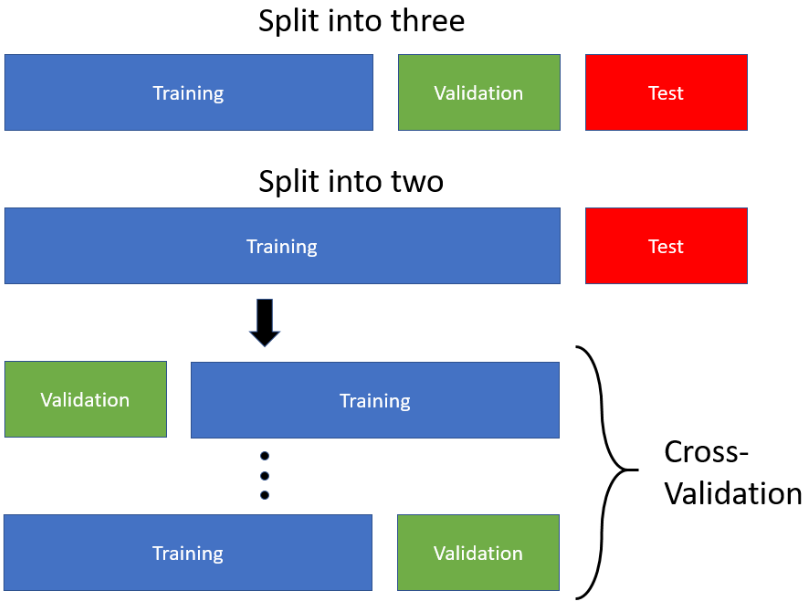


Figure 3.6: Illustration of how Cross-Validation divides the data into a training and a validation set

The choice of K is not straight forward. Choosing a low K, say $K = 2$, we get a low variance but the bias can be high depending on how the performance of the model varies with training size. The benefit of having a low K is that it is computationally cheap. Choosing K between 5 and 10 are common practices and considered as a good compromise (Kohavi, 1995) and (Breiman and Spector, 1992). With $K = m$, known as leave-one-out cross-validation, the estimate of Err is approximately unbiased but can have high variance. Also, the computational cost can be very high. Leave-one-out cross-validation is considered an exhaustive method as it learns and tests in all possible ways to divide the original sample into a training and a validation set.

3.5.3 Goodness of Fit

The goodness of fit of a statistical model describes how well it fits with a set of observations. A goodness of fit measure usually summarize the differences between observed values and values given by the model in question. For regression models, there are several measures to assess the goodness of a fit. These measures are generally divided into two categories:

- How well the model fits the training data.
- How well the model fits new, unseen test data.

To calculate the differences between observed values and values given by the model we normally use a loss function as previously explained, or we can use cross-validation to estimate the prediction error. In words, we are measuring the error of the model, and the test error over an independent test set can be compared with a base error rate. However, we also want to assess how well our model fits the training data. A typical measure of the goodness of fit is the coefficient of determination, also known as R^2, or extensions of it such as *adjusted* R^2. These measures are only valid for linear regression models and we need a measure to compare non-linear methods such as nearest-neighbors to our linear models. We also want to take the model complexity into account. One such measure is the Akaike information criterion (AIC), introduced by Akaike (1974). AIC is derived from the log-likelihood function and offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. Doing this, AIC provides a compromise between the goodness of fit of a model and the model complexity.

3.5.4 Variable Selection

We are often faced with the problem of variable selection in machine learning and statistical learning. The problem is how to select a subset of the input variables that best explains the variability or best reduces the prediction error, in our target value. There are mainly four reasons we use variable selection:

- simplification of models to make them easier to interpret (Hastie et al., 2001),
- shorter training times,
- to avoid the curse of dimensionality (Bellman and Roth, 1986),
- enhanced generalization by reducing overfitting (reduction of variance).

A central premise when using variable selection is that the input data contains variables that are either redundant or irrelevant and can thus be removed without incurring much loss of information (Bermingham et al., 2015). The variable selection process is no different from model selection based on parameter tuning. However, as the number of variables m get sufficiently large, the number of possible subsets increases exponentially as $2^m - 1$. This causes algorithms that check all possible combinations to become computationally exhaustive.

Especially in linear regression one may want a simple model with few variables for easy interpretation of the results. Again we are faced with the bias-variance tradeoff. We want a good model in terms of generalization and goodness of fit while keeping the complexity of our model low to avoiding overfitting.

Backward-stepwise selection is a greedy algorithm that starts with all input variables. For each iteration, the algorithm removes the variables that decrease the fit of the model the least. This means that we remove the variables that add the least to the residual sum of squares. This process leads to a sub-optimal search through all the possible subset of variables. However, this method is often preferred due to the computational efficiency compared to an exhaustive search through all possible combinations. Since the residual sum of squares will increase for each variable added, we need some other criterion to choose the optimal subset size k. Hastie et al. (2001) discourages the use of F-statistics to select variables based on their significance and proposes the use of cross-validation as a way to choose the optimal subset size k. Forward-stepwise selection is a similar approach to the backward-stepwise selection. In this algorithm, you start with zero variables, only the intercept, and add the variables that best improves the fit of the model.

As the stepwise selection methods do only test a sub-optimal set of all possible combinations they are often discouraged to use. Both Briggs (2008) and Flom and Cassell (2007) address known problems with the stepwise methods, discourage the use of level of significance to add or drop variables and stress the importance of cross-validation and using independent test sets.

3.5.5 Example

To provide an example of cross-validation, least-squares regression and the interaction between bias and variance error we generate data according to Equation 3.30.

$$y = x + 3\sin(x) + \epsilon \tag{3.30}$$

where ϵ is normally distributed noise with zero mean and a variance of 2, that is $\epsilon \sim N(0,2)$. We generate 200 points using this function with x-values between 0 and 20. We divide the generated data randomly into a training set of 80 % and an independent test set of 20 %. The training and test data can be seen in Figure 3.7.

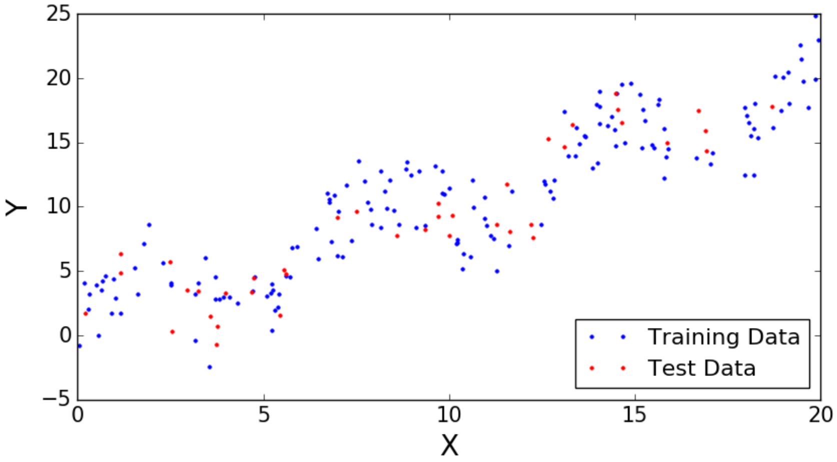


Figure 3.7: Data generated for regression and cross-validation example

We assume a linear regression model where the inputs variables $\mathbf{x}_j$ are basis-expansions of the quantitative input $\mathbf{x}_1$. That is $\mathbf{x}_j = \mathbf{x}_1^j$. If we let α denote the highest order of polynomials to include in our model, α will correspond to our model complexity. We want to find the optimal model $\hat{f}_\alpha$ that minimizes the estimated test error Err.

We will use 5-folds in our cross-validation for each α from 0 (constant model) to 15. The MSE is used as the loss function. The estimated training error $\mathrm{E}[\overline{\mathrm{err}}]$ and estimated test error Err as a function of α can be seen in Figure 3.8.

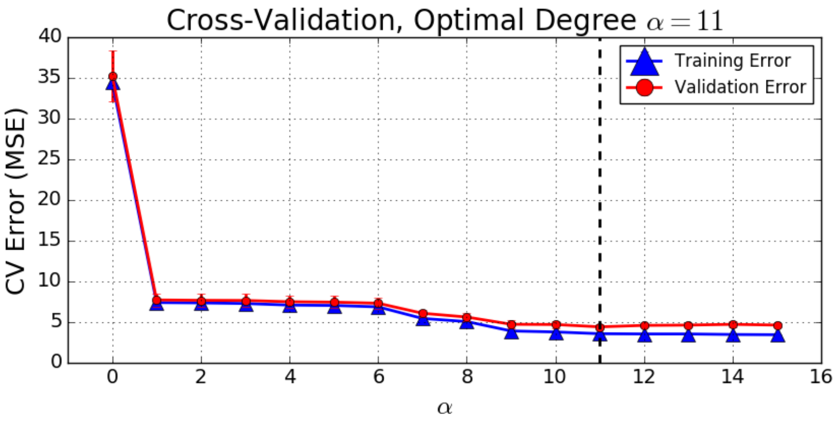


Figure 3.8: 5-fold cross-validation used on polynomial regression for polynomial of degree 0 to 15.

When $\alpha = 0$ we have a constant model and only one parameter β_0. With a constant model there is no variability in the model predictions and the expected values of our predictions are far off the target values. This gives us a high bias and a low variance, resulting in underfitting a high expected prediction error Err. When $\alpha = 15$ we have a polynomial of degree 15 and 16 parameters to estimate. We now have a high variability in the model predictions around their means. This comes at the cost that the expected values of a prediction is closer to its target value, and we get a low bias but a high variance. Thus, we get an overfitting and a higher expected prediction error Err. The expected training error $\mathrm{E}[\overline{\mathrm{err}}]$ decreases as a function of α as expected. We see that the optimal complexity of our model is $\alpha = 11$.

In Figure 3.9 three polynomial regression curves are fitted to generated data for $\alpha = 0, 11, 15$ to illustrate the differences between underfitting, overfitting and the optimal model. It is clear that for $\alpha = 0$ the model fails to predict the values from $\mathbf{x}_1$. For $\alpha = 15$ the model fits the variability in the target values due to noise. When $\alpha = 11$ represents the model with lowest expected test error and thus is chosen as the optimal model.

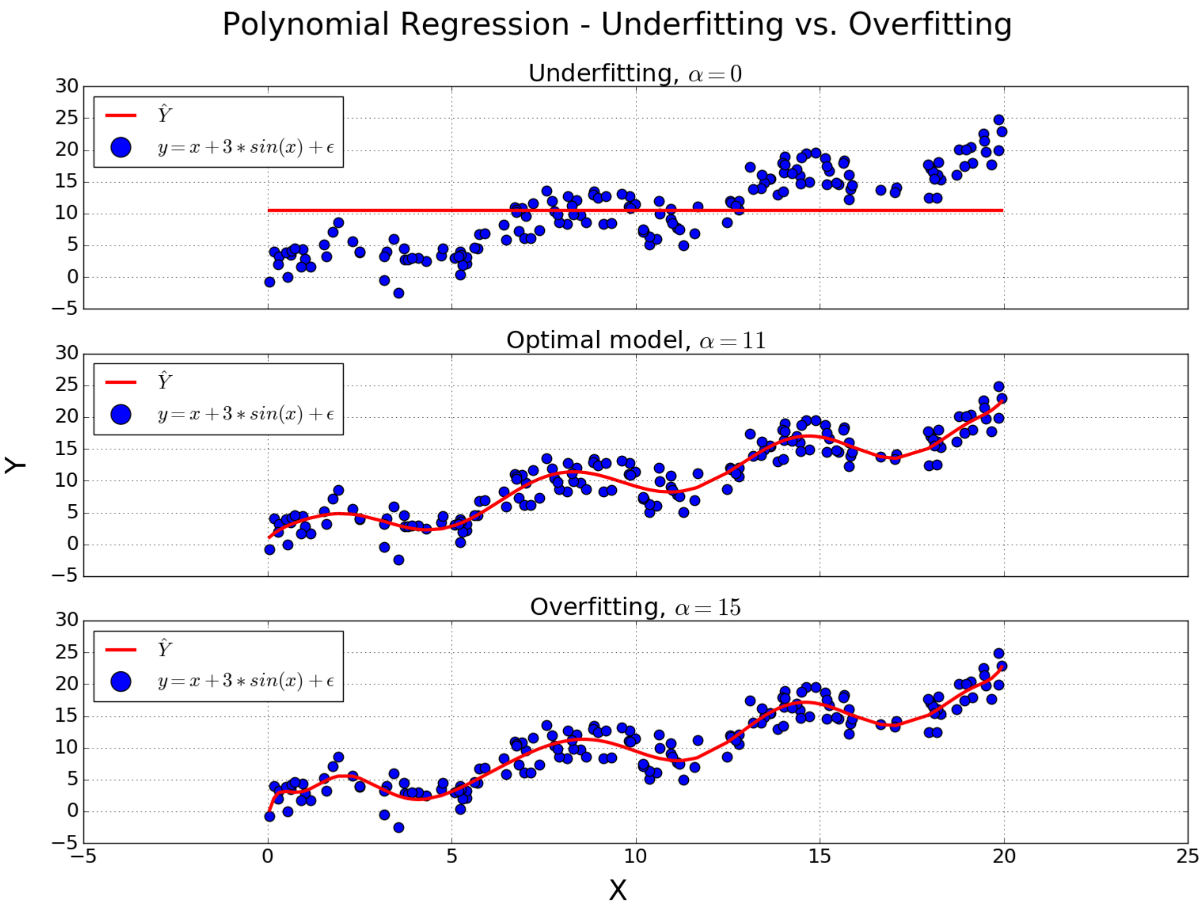


Figure 3.9: Polynomial regression curves for three different $\alpha = 0, 11, 15$.

Chapter 4

Database Preparation

4.1 Introduction

In this Chapter we prepare or preprocess the given data into a dataset suitable for analysis. The given data from DNV GL is already preprocessed to some degree by the storage and logging system on the vessel, we have no information on how this is done and what has been done to the data before we retrieve it. In Section 4.2 we briefly go into how the data was handed to us and how we store the data to make it more accessible and easier to work with. To remove obvious outliers we apply simple filters, this is done in Section 4.3. In Section 4.4 we define a method to handle the difference in temporal resolution of the variables. In Section 4.5 we redefine multiple variables. The fully preprocessed dataset is shown in Section 4.6. Finally, in Section 4.7, we use unsupervised learning to investigate the dataset made from the preprocessing.

4.2 Data Storage

The data we got from DNV GL were stored in an inefficient manner for doing analysis. Due to this, the first objective was to transfer the data to a storage format that would be more efficient for further preprocessing and analysis then the original comma separated values (CSV) format. The data from DNV GL were transformed to Hierarchical Data Format (HDF) which is faster to load and takes less space to store large amounts of data (HDFGroup, 2017).

When all files were converted to HDF format, each variable for each 6-hour file was saved in one file, giving us a total of 25 files, one for each variable. Saving the data this way made it easy to investigate each variable separately and only open the variables of interest. This gave an easy to use and efficient storage system and further preprocessing and analysis could be carried out without much hassle.

4.3 Data Reduction

When inspecting the unfiltered data it was obvious outliers were present in the data. An example of this can be seen in Figure 4.1 where we see water depths logged of approximately −16000 [m], deeper than the deepest point on earth.

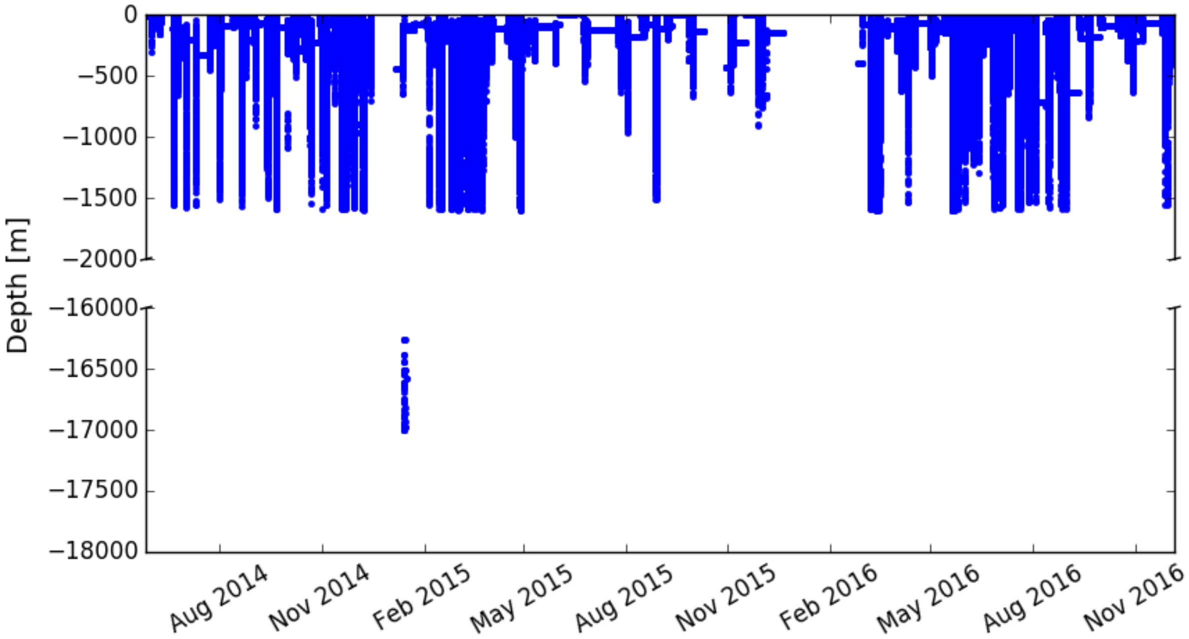


Figure 4.1: Unfiltered water depth data with obvious outliers around January 2015

These obvious outliers can be found in several other variables as well. To cope with this, simple minimum and maximum filters were added to each variable according to Table C.1 in Appendix C.

From *speed through water* there were removed 684 observations, the reason can be seen in Figure 4.2. Here we see points reaching as high as 1000 [kts] around February 2015.

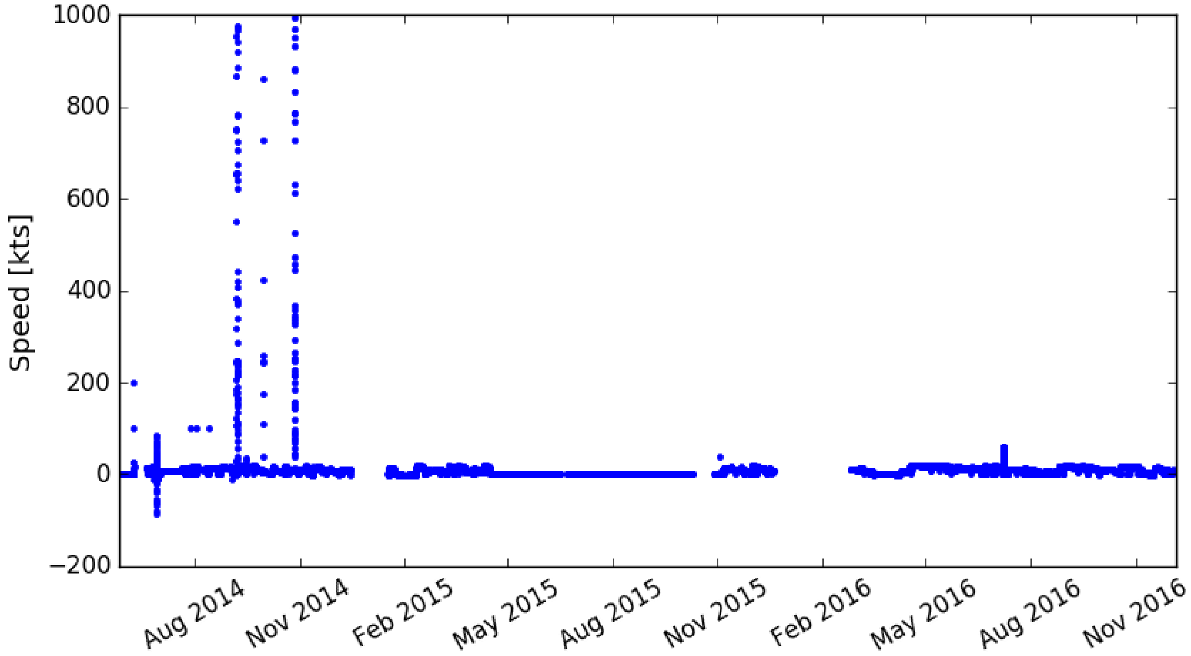


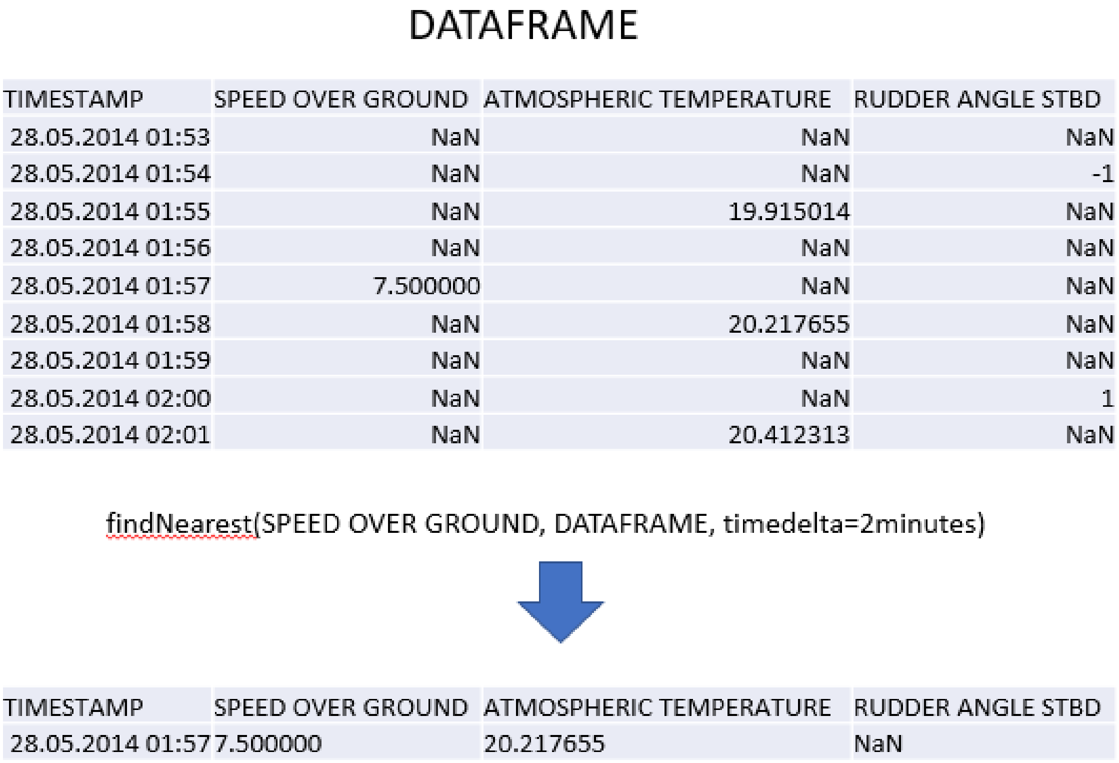
Figure 4.2: Unfiltered speed through water data with obvious outliers around from August to November 2014

The data is also filtered such that other losses then the R_{AH} can be assumed negligible. This includes removing incidents where the wind speed is above 4 on the Beaufort scale (BF) (above 16 [kts]). When the wind speed is BF 4 or lower, the waves are assumed to be less than 2 meters high, and thus we can also neglect the wave resistance. The Beaufort Scale can be seen in Appendix D.2. The quality of the wind data is poor as we saw in Figure 2.4 in Section 2.3.1. To filter out the conditions with BF 4 or higher, we use the GPS position of the vessel matched up reanalysis data from National Center for Environmental Information (NOAA, 2017).

Finally, a nearest neighbor outlier removal is run set to remove 10 % of the points that are furthest away from its neighbors. The *Performance Loss* plotted before and after the nearest neighbor outlier removal can be seen in C.2. We clearly see that some of the most obvious outliers are removed.

4.4 Find Nearest

All sensors are logging at different timestamps and variables are rarely logged at the exact same timestamp. To cope with this we find the values closest to a given timestamp with a maximum allowed time delta t_Δ. The maximum allowed time delta is added to ensure that values are not several hours away from the timestamp we are interested in. A small example to visualize this process can be seen in Figure 4.3. Here speed over ground is used as the main variable such that all the other variables are matched to the timestamps of speed over ground. We add Not-a-Number (NaN) if there are no values inside the given t_Δ.



DATAFRAME

TIMESTAMP	SPEED OVER GROUND	ATMOSPHERIC TEMPERATURE	RUDDER ANGLE STBD
28.05.2014 01:53	NaN	NaN	NaN
28.05.2014 01:54	NaN	NaN	-1
28.05.2014 01:55	NaN	19.915014	NaN
28.05.2014 01:56	NaN	NaN	NaN
28.05.2014 01:57	7.500000	NaN	NaN
28.05.2014 01:58	NaN	20.217655	NaN
28.05.2014 01:59	NaN	NaN	NaN
28.05.2014 02:00	NaN	NaN	1
28.05.2014 02:01	NaN	20.412313	NaN

findNearest(SPEED OVER GROUND, DATAFRAME, timedelta=2minutes)

TIMESTAMP	SPEED OVER GROUND	ATMOSPHERIC TEMPERATURE	RUDDER ANGLE STBD
28.05.2014 01:57	7.500000	20.217655	NaN

Figure 4.3: A visualization of the find nearest function. Speed over ground is used as the main variable and a time delta of two minutes is used

If we do this process for all timestamps in one variable and remove the rows with any occurrence of NaN we would end up with a full dataframe or matrix. The size of the table would depend on the t_Δ sent to the find nearest function. In Table 4.1 we see the remaining number of data observations when running the find nearest process on speed over ground with 5 different t_Δ. As speed over ground has relatively few measurements (7566) it is expected that we end up with quite a few points after running the find nearest function. However, when we run this function, we are sure that all other variables are logged within $\pm t_\Delta$ for each timestamp. By doing this we assume that the variables do not change much within $\pm t_\Delta$ of each timestamp. For a vessel of this size, this might be a fair approximation most of the time for $t_\Delta < 10$ [min].

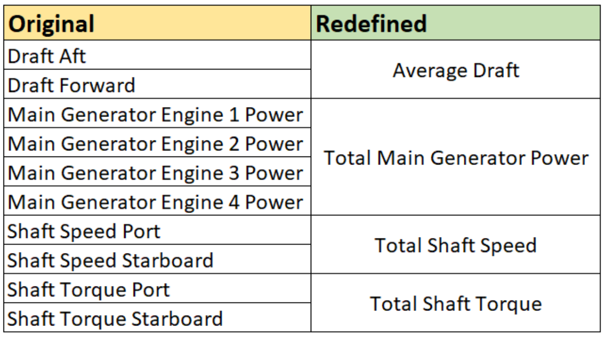
t_Δ **in minutes**	**Size of remaining dataframe**
60	4792
30	4395
10	4062
5	3683
1	2780

Table 4.1: Find nearest function for different timedeltas with speed over ground as main variable

Speed over ground is used as the main variable as this is arguably the most important variable when inspecting the performance loss. Ideally, speed through water should have been used, but these measurements were unreliable most of the time as we saw in Figure 4.2.

4.5 Variable Redefinition

Several of the chosen variables essentially measure the same metric for similar subsystems, such as the shaft torque port and starboard, assuming the vessel is not using the difference in starboard and port propeller to turn. In this analysis, we are not interested in the individual behavior of these subsystems and thus redefine them as one variable by means of summation or remove the variable. The list of redefined variables is shown in Figure 4.4. This reduces the amount variables from 25 to 13. For a more in-depth analysis, the individual variables should be kept separate if one wants to investigate the differences between them.



Original	Redefined
Draft Aft	Average Draft
Draft Forward	
Main Generator Engine 1 Power	Total Main Generator Power
Main Generator Engine 2 Power	
Main Generator Engine 3 Power	
Main Generator Engine 4 Power	
Shaft Speed Port	Total Shaft Speed
Shaft Speed Starboard	
Shaft Torque Port	Total Shaft Torque
Shaft Torque Starboard	

Figure 4.4: List of variable redefinition by summation.

To make use of the dry-dock and propeller polishing dates, we add them as variables, *Days since dry-dock* and *Days since propeller polishing*. This way any temporal changes and drastic changes due to dry-dock or propeller polishing might be captured by our regression models.

4.6 Prepared Database

After all the preprocessing steps described in this chapter are done, we are left with what we from now will refer to as the *prepared dataset*. The prepared dataset contains 13 variables and 1833 observations. All the variables and their unit can be seen in Table 4.2. The *Percentage Speed Loss* (performance loss) will be our target value and the remaining 12 variables will be our input variables in the regression analysis. All input variables are normalized.

Variable	**Unit**
Atmospheric Temperature	C
Speed Over Ground	kts
Speed Through Water	kts
Sea Water Temperature	C
Water Depth	m
Shaft Power Total	kW
Total Main Generator Power	kW
Draft Average	m
Days Since Dry-Dock	days
Days Since Propeller Polish	days
Total Shaft Torque	kNm
Total Shaft Speed	rmp
Performance Loss	%

Table 4.2: Variables in prepared dataset.

4.7 Date Exploration

We now want to explore the prepared dataset using the unsupervised method described in Chapter 3. There are several correlations and patterns in the dataset we expect to see. If these correlations and patterns do not show up, it gives an indication the information was lost during the preprocessing.

4.7.1 Correlation Matrix

Figure 4.5 shows the correlation matrix for the 13 variables. Such a matrix simply presents the variable-variable correlations ranging from -1 (completely negatively correlated) to 1 (completely positively correlated). This is a simple method to get an overview of the correlations between variables or variable clusters. We expect to see correlations between variables such as speed over ground and speed through water. This gives an indication that the processed data still contains relevant and correct information. If the correlations in the matrix make sense, it gives an indication that we have not lost much information during the preprocessing.

We clearly see that the parameters directly connected to the propulsion system, like *shaft torque*, *shaft speed* and *speed over ground*, are highly positively correlated which is expected. The *sea water temperature* and *atmospheric temperature* are highly correlated which is also expected. We are also pleased to see that variables that have no logical relations are insignificantly correlated.

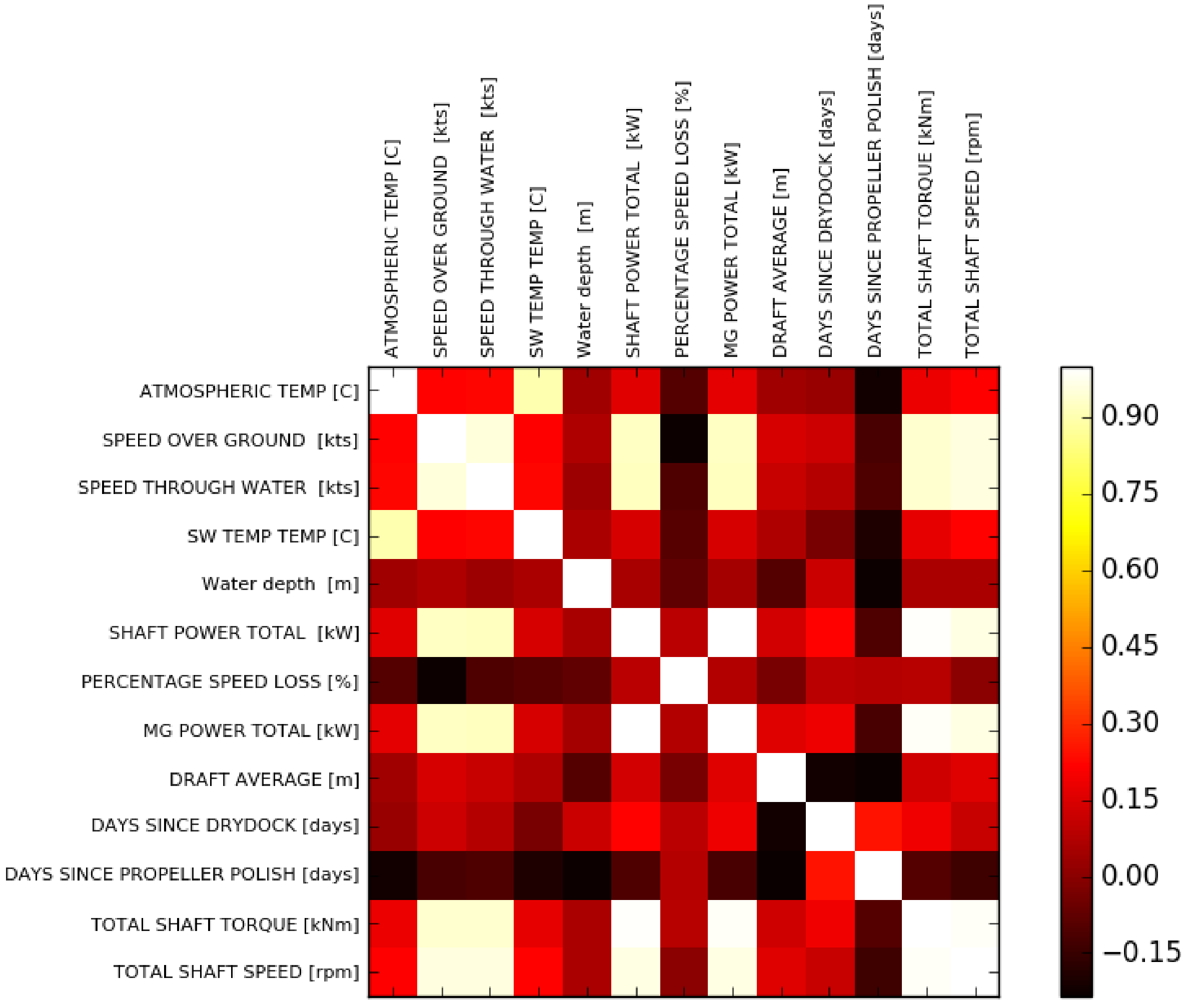


Figure 4.5: Correlation matrix for the chosen 13 variables

4.7.2 Principal Component Analysis

With the prepared dataset, we use the previously described PCA method (Section 3.3.2) to reveal the structure in the data and investigate which variables best describe the variation in the data. It will also indicate how the different variables are correlated similarly to the correlation matrix in Section 4.7.1, but with a clearer representation.

4.7.2.1 Explained Variance

In Figure 4.6 we see how the explained variance develops depending on the number or principal components (PCs) included. The explained variance starts out quite low at 46 % but already increases to 94 % with 6 PCs. The contribution of explained variance by adding an additional component declines in an exponential fashion. Furthermore, the interpretation of the higher components become less and less intuitive.

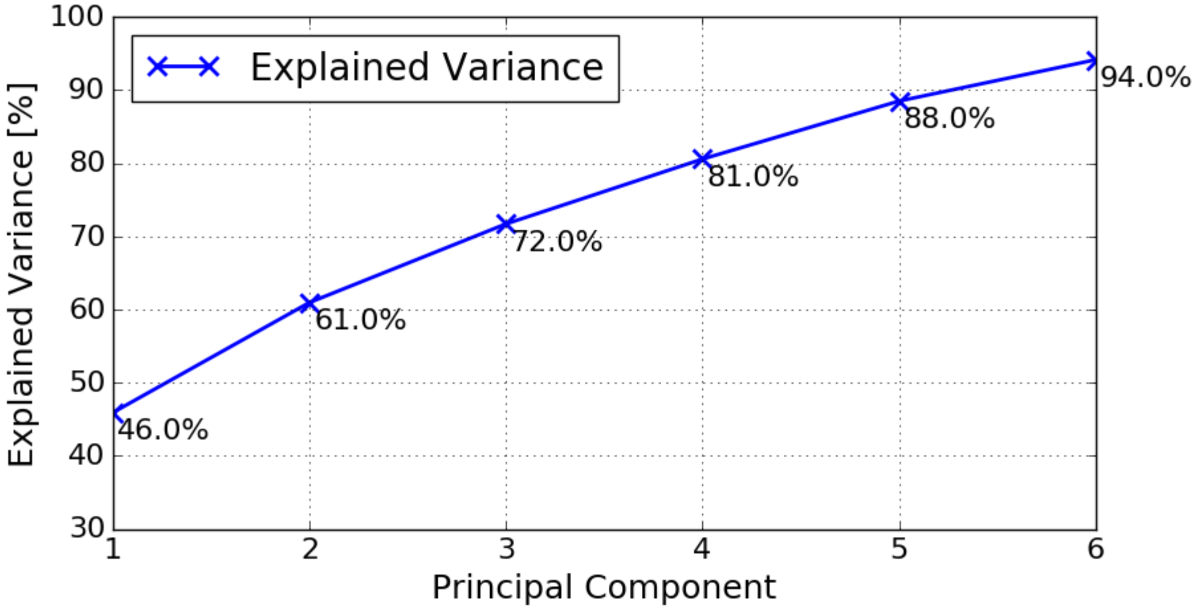


Figure 4.6: Cumulative explained variance after 6 PCs.

4.7.2.2 Scores and Loadings

It is important to view the scores plot in combination with the loadings plot to get an understanding of the scores plot and to see which variables influence the different PCs. Only the fist two components are included in the scores plot as the higher PCs describe less variance and there is no clear pattern in these score plots.

The first two components describe the most of the variance in the data, in this case, they describe 61 % of the variance. It is in these two components we expect to see most of the already known or expected correlations. Variables that are positively correlated are located close to each other while the negatively correlated variables are found on opposite sides of each other. Hence, variables on opposite diagonals will be negatively correlated in both PCs.

In Figure 4.7 the loadings for PC1 and PC2 are plotted together. We start by noticing the variables close to the origin of the loadings plot, such as the *percentage speed loss*. These variables describe little of the variance in these first two PCs. Notice the cluster of points to the right in the loadings figure. In this cluster, all the variables related to propulsion is located. This means that the propulsion variables are highly correlated and this is expected. We also notice that *atmospheric temperature* and *sea water temperature* does not describe much variance in the first component, but describes much of the variance in the second component.

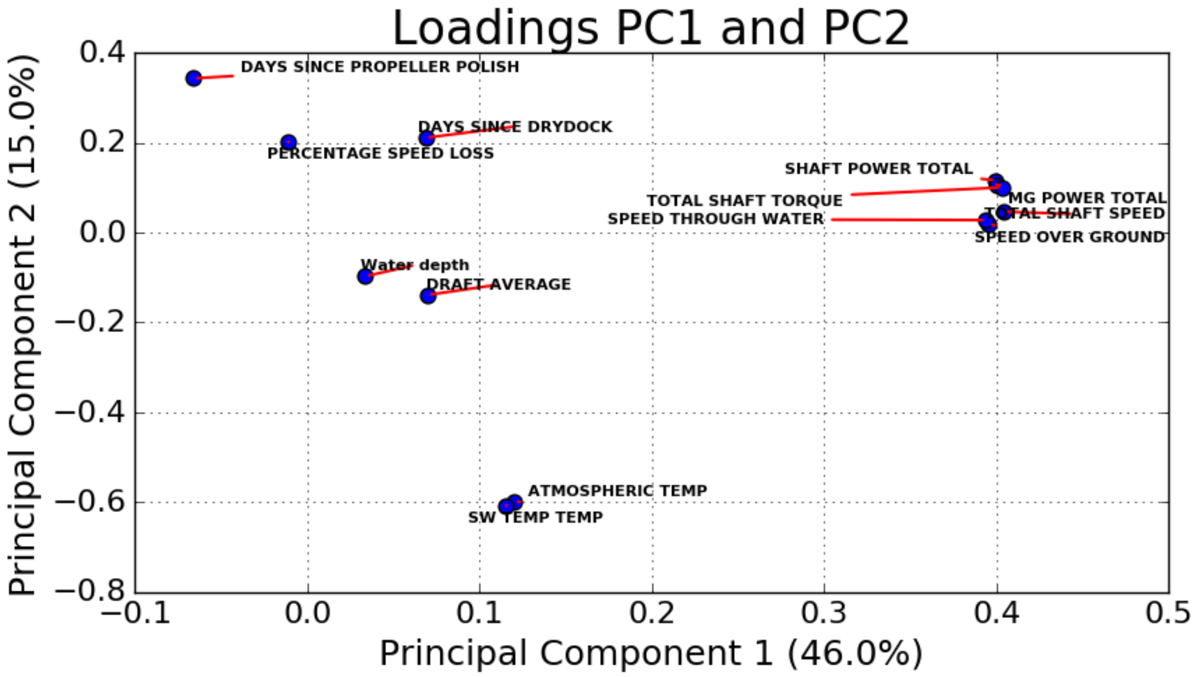


Figure 4.7: Variable loadings for PC1 and PC2.

In Figure 4.8 the scores for PC1 and PC2 can be seen. Each of the four plots has been color coded to the four variables *speed over ground*, *water depth*, *atmospheric temperature* and *average draft*. By coloring the scores according to various variables we can easily see how the different variables are affecting each principal component. Coloring by *water depth* is done to visualize that water depth does not describe much variance in the first two components.

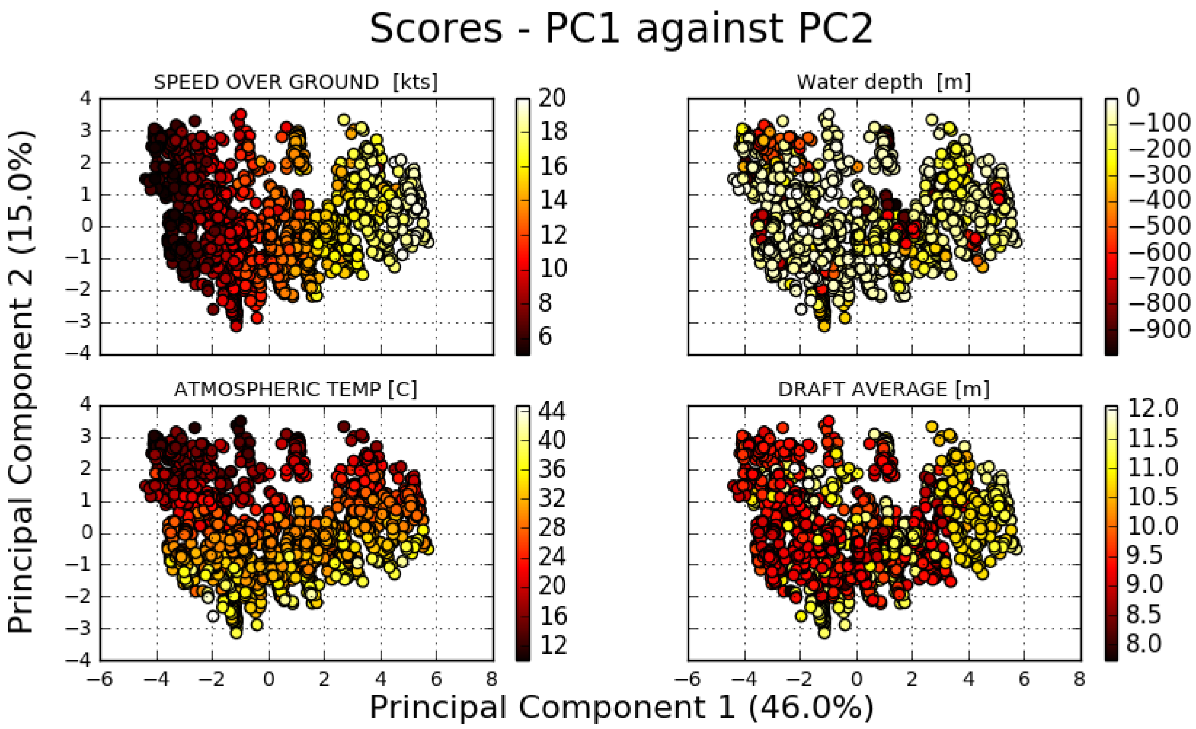


Figure 4.8: Variable scores for PC1 and PC2 color coded by four different variables as indicated by the title for each plot.

An alternative representation of the PCs and its loadings can be seen in Figure 4.9, inspired by Perera and Mo (2016). Here each variable loading is presented for the first six PCs in a two-dimensional grid. The magnitudes of the loadings are proportional to the area of the circles and the color of the circles represents the signed magnitudes of the loadings. This figure also represents an overview of the correlations among the respective parameters of ship performance and navigation information.

Presenting the variable loadings for each PC in this manner one can easily visualize the variables with higher loadings and their positive and negative correlations, resulting in the same analysis as above for the first two components. We see that *percentage speed loss* describes much of the variance in PC 5. In this PC, the other variables are quite small, indicating that the *percentage speed loss* is not strongly correlated with any variable. This tells us that predicting the *percentage speed loss* might be hard as there is no clear relation with the other variables.

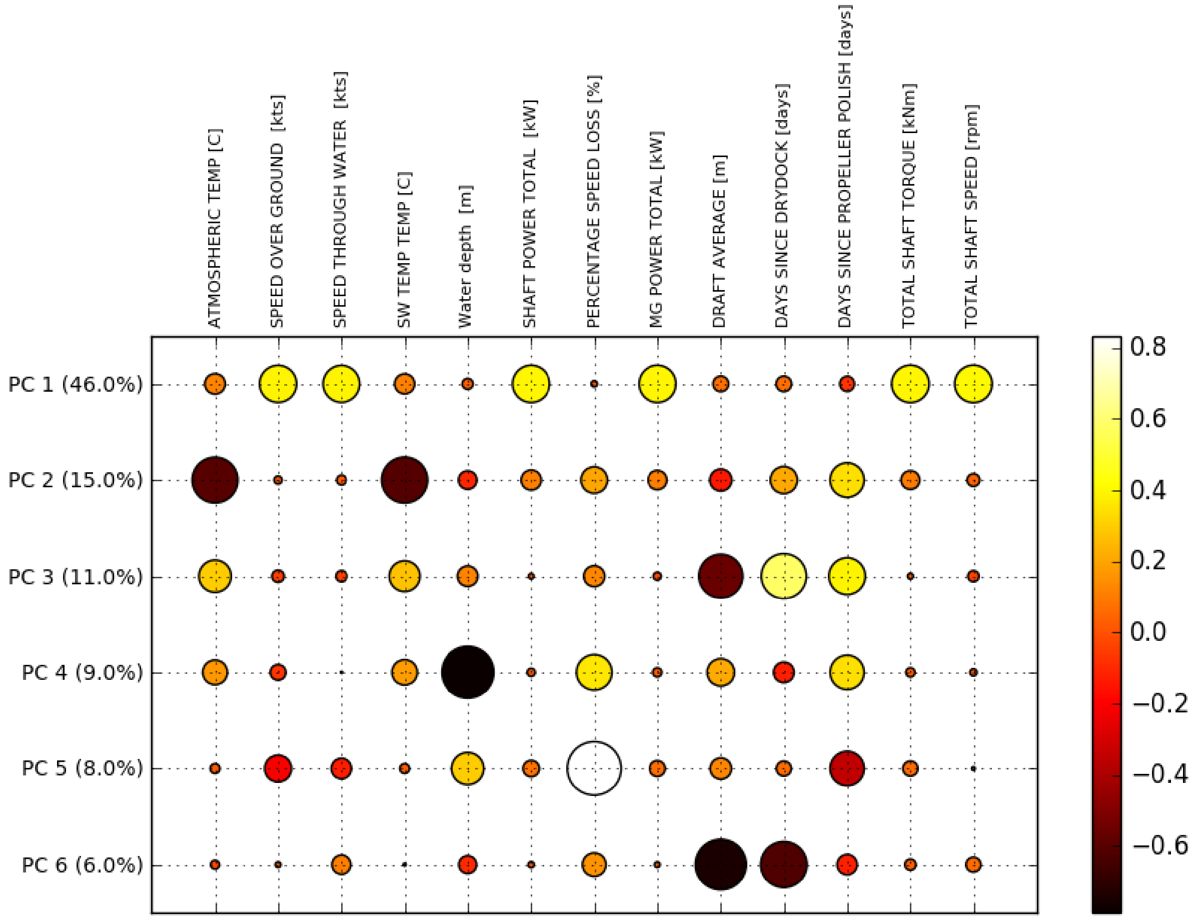


Figure 4.9: Variable loadings for the 6 principal components.

Chapter 5

Analysis

In this chapter, various methods from Chapter 3 will be used to explore the underlying relationship between the vessel performance and the 12 chosen variables. An extra effort will be put into analyzing the vessel performance relation with time (hull-propeller performance). The chapter will be divided into two main parts: In the first part, Section 5.1, we simulate data to mimic the real data. A methodology for regression will be employed to recover the relationships in the simulated data. In the second part, Section 5.2, the same methodology is applied to real-world data. In this way, we are better able to explain why or how the real-world analysis fails to display the expected results.

5.1 Simulated Data

We now assume three situations where the performance loss due to hull-propeller performance can be explained as a function of virtual time t_v, that is $\mathrm{L} = f(t_v)$. The function $f(t_v)$ will be defined for three cases:

1. Linear model, the loss is a linear function of virtual time, and virtual time is equal to real time. This means that one day in real time is equal to one day in virtual time, $t_v = t_r$.

2. Non-linear model, the loss is a non-linear function of virtual time. The virtual time is still equal to real time, $t_v = t_r$.

3. Multidimensional non-linear model, the loss is a non-linear function of the virtual time, and the virtual time is dependent on various variables. That is, $t_v = f(P)$ where

P is a vector of variables. The variables in P can, for instance, be the sea water temperature and real time.

Noise will be added to all models, and a nearest-neighbor outlier removal algorithm will be used to remove 10 % of the points, as was done for the real-world data. When we have generated data from our simulated models, we will perform the methods described in Sections 3.4 and 3.5 to verify that the methods are able to uncover the models used to generate the simulated data. For all simulated models, time (both t_r and t_v) is counted in days.

5.1.1 Linear Model

In this case, we simply assume that the speed loss of the vessel is a linear function of real time and that virtual time is equal to real time such that $t_v = t_r$. The speed loss function can then be defined as

$$\mathrm{L} = a t_v + \mathrm{L}_0 + \epsilon \tag{5.1}$$

where L is the speed loss in percentage, $\mathrm{L}_0 = 5$ is the initial loss at $t_v = 0$, $a = \frac{100}{3650} = 0.0274$ (gives a loss of 30 % over approximately three years) and $\epsilon \sim N(0,4)$. We then generate data for 1200 days and divide them randomly into a test set of 20 % and a training set of 80 % as seen in Figure 5.1.

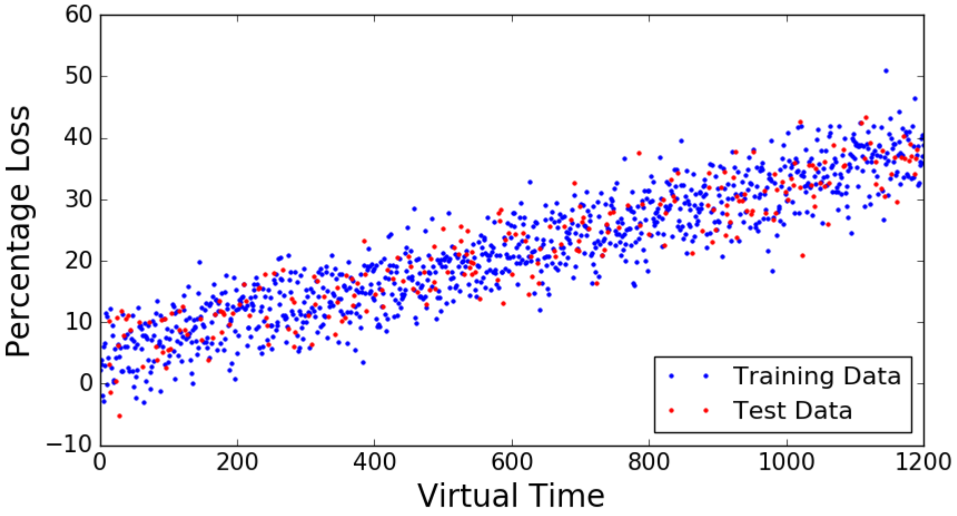


Figure 5.1: Data generated from linear model with added noise and divided into a training and a test set.

Using this simple linear model, we assume that time is the only parameter that affects the speed loss. There are no events that affect the speed loss such as propeller polishing or damaging of the propeller. Linear least squares regression, polynomial least squares regression, and nearest neighbor regression will now be used on this linear model to test these methods on data which we know the underlying equations.

5.1.1.1 Linear Least Squares Regression

We start out by testing linear least-squares regression. Since the data is generated from a linear model, the linear regression is expected to find regression coefficients β_0 and β_1 similar to L_0 and a. However, they might be slightly off due to the added noise. In Figure 5.2 the linear regression line is plotted together with the generated data.

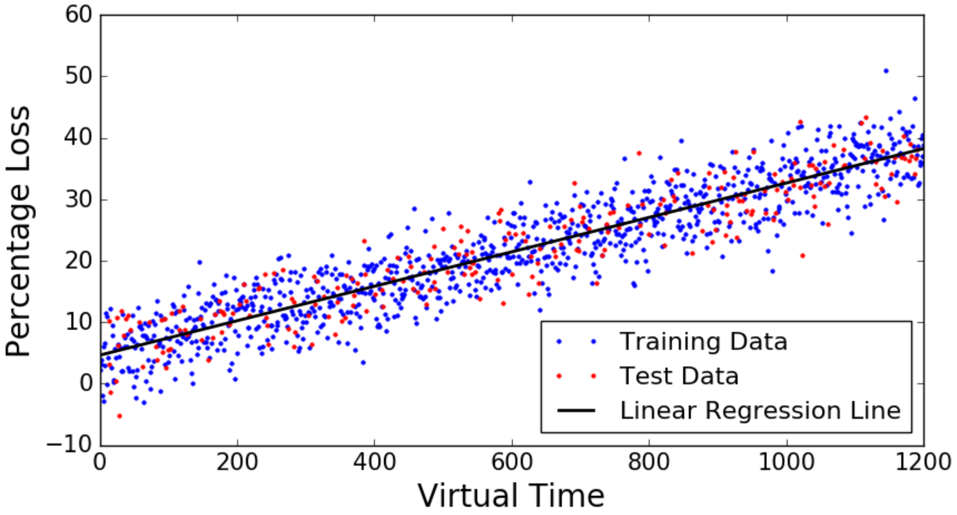


Figure 5.2: Data generated from linear model together with the linear regression line generated from least squares regression.

This regression line has the coefficients $\beta_0 = 4.61$ and $\beta_1 = 0.028$. These coefficients are similar to those coefficients used to generate the data ($L_0 = 5$ and $a = 0.027$), we see that the linear regression method is well suited to find relations from linear interactions, as expected. The base error is reduced by 86.5 %.

5.1.1.2 Polynomial Least Squares Regression

We now want to use polynomial regression in combination with cross-validation to find an optimal degree of polynomials to include in the regression. As mentioned in Section 3.4.1, polynomial regression is still linear in that the inputs can be basis-expansions of quantitative inputs, such as $\mathbf{x}_2 = \mathbf{x}_1^2$. As the model used to generate the data is linear, it is expected that we might get the best prediction error by only using a polynomial of degree one. If this is the case, the regression coefficients will be the same as the previously tested linear regression line. However, due to the noise added, we might also get that polynomials of higher degrees get a better estimated prediction error. We use polynomials from degree 0 to 7 such that the regression model looks like

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{\alpha} \mathbf{x}_j \beta_j, \tag{5.2}$$

where $\alpha \in 0, 1, \cdots, 7$ and $\mathbf{x}_j = \mathbf{x}^j$. We will then perform a 5-fold cross validation for each of the 8 cases to find the optimal degree for α. The case when $\alpha = 0$ means taking the average of the data and use this as a prediction for new values. This is the worst prediction model and is often referred to as the base error. The cross-validation process can be seen at the top of Figure 5.3.

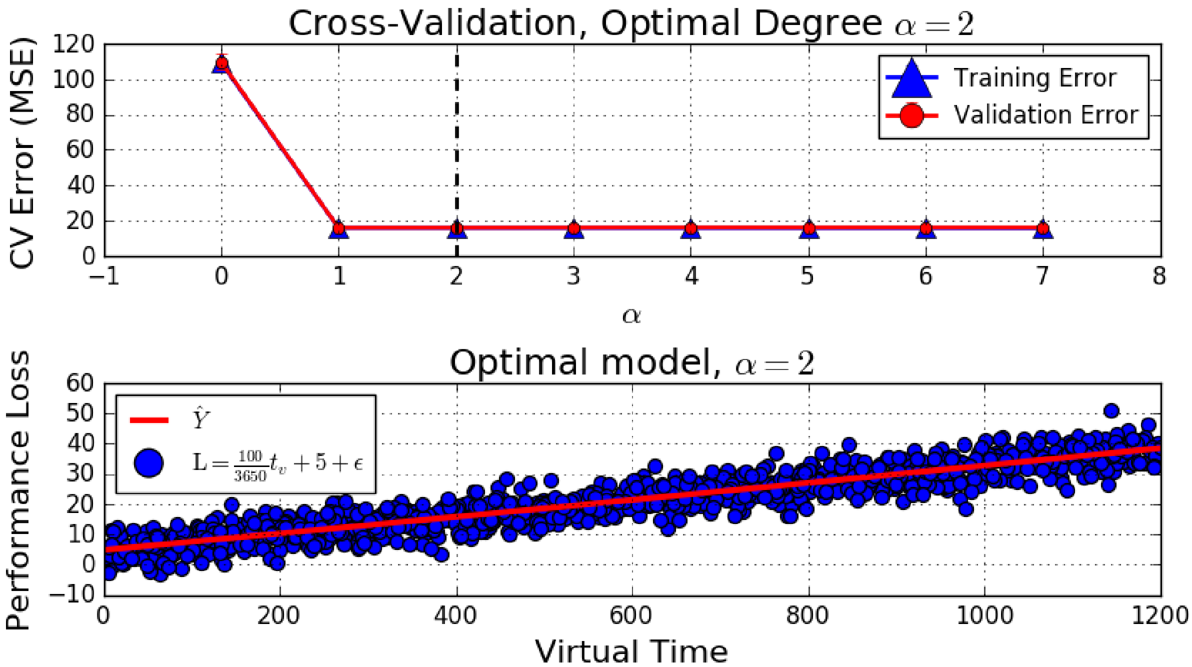


Figure 5.3: Top: Cross Validation to find optimal polynomial degree. Vertical black-dotted line represents the optiaml degree. Bottom: Polynomial regression line for optimal degree.

As we see in Figure 5.3 a polynomial of degree $\alpha = 2$ gives the lowest estimated test error. The regression line for $\alpha = 2$ can be seen at the bottom of the figure.

α	**Estimated Test Error** (Err)	**Standard Deviation**
0	109.82	4.30
1	15.889	0.058
2	15.887	0.064
3	15.952	0.091
4	16.040	0.124
5	16.039	0.114
6	16.015	0.115
7	15.990	0.122

Table 5.1: Estimated test error and standard deviation for polynomial regression on linear model for different α

In Table 5.1 we see that the error for $\alpha = 1,2,\ldots,7$ does not differ much and they all have low standard deviations. Since the error when $\alpha = 1$ is within one standard deviation from $\alpha = 2$ we would choose a polynomial of degree 1. This makes the regression model easier to interpret. By choosing $\alpha = 1$ the test error on the independent test set $\text{Err}_{\mathcal{T}}$ becomes 14.86, and the base error is reduced by 86.5 %, same as the linear least squares regression. We notice that $\text{Err}_{\mathcal{T}} = 14.86$ is lower than the expected test error $\text{Err} = 15.899$. This is due to the relatively low number of folds $(K = 5)$ in the cross-validation and how the data is randomly divided into a test and training set.

5.1.1.3 Nearest-Neighbor Regression

We now use the uniform weighted nearest-neighbor regression described in Section 3.4.2.1. To find the optimal number of neighbors we use a 5-fold cross-validation. This process is seen in the top of Figure 5.4. We have tested for neighbors $k = 1,2,\cdots,70$. For nearest-neighbor regression it is hard to predict which k will give the optimal fit. Since the model used to generate the data is quite simple, it is expected that the number of neighbors should be quite high, as a large k gives a simpler model then a small k.

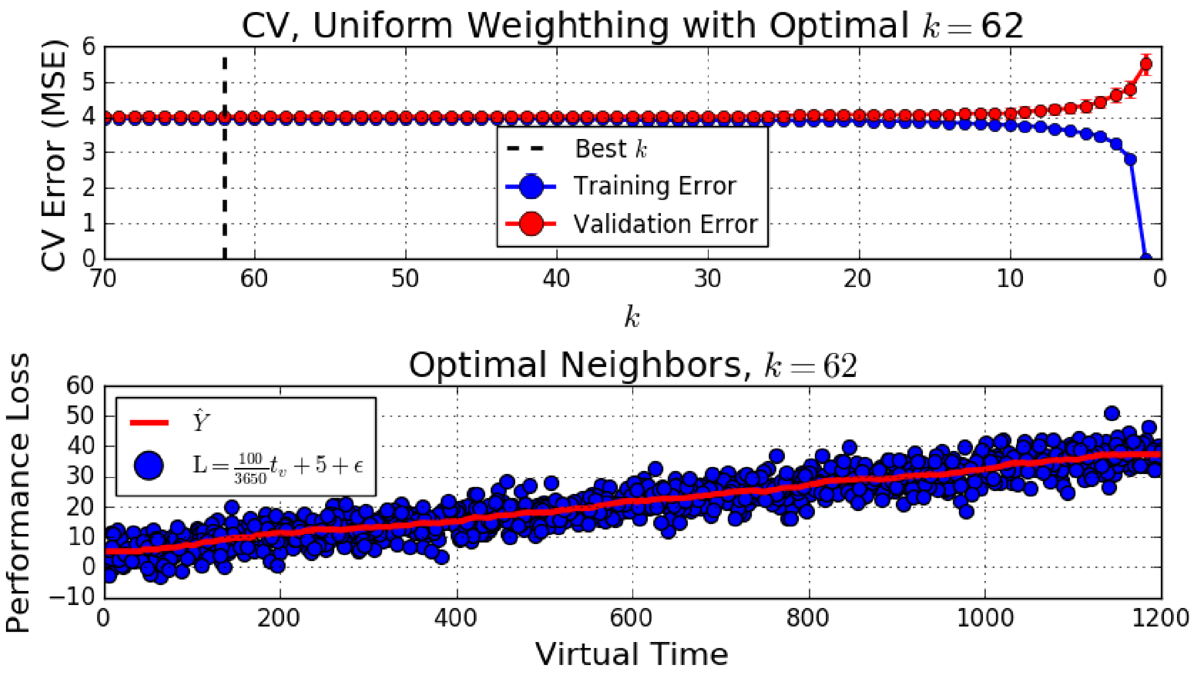


Figure 5.4: Top: Cross Validation to find optimal number of neighbors. Vertical black-dotted line represents the optimal k. Bottom: nearest-neighbor regression line for optimal k.

From the figure we see that the optimal number of neighbors $k = 62$ is quite high as expected. There is only a small change in MSE when $k > 20$. In the bottom of the figure the regression line is plotted for $k = 62$, this regression model gives $\mathrm{Err}_{\mathcal{T}} = 14.94$ on the independent test set. This is a base error reduction of 86.4, very similar to the linear regression methods.

5.1.2 Non-Linear Model

In this case we assume that the speed loss of the vessel is a non-linear function of virtual time t_v and that virtual time is equal to real time, $t_r = t_v$. We assume that the development of the added resistance on the vessel (the performance loss) is given as assumed by Gundermann and Dirksen (2016)

$$R_{AH} = A \tanh(B t_r) \tag{5.3}$$

where A is the asymptotic amplitude, B is the growth rate and t_r is the time. The coefficients A and B depends on the increments of t_r (whether it is in seconds, hours, months or years). If we further assume that time is the only effect on speed loss of the vessel we can make a non-linear loss function as

$$\mathrm{L} = A\tanh(Bt_v) + \mathrm{L}_0 + \epsilon \tag{5.4}$$

where L is the speed loss in percentage, $\mathrm{L}_0 = 5$ is the initial loss at $t_v = 0$, A and B is chosen to 30 and 0.0018 respectively. This gives a speed loss of 35 % as $t_v \to \inf$. The noise is assumed gaussian, $\epsilon \sim N(0,4)$. We then generate data for 1200 days and divide them randomly into a test set of 20 % and a training set of 80 % as seen in Figure 5.5.

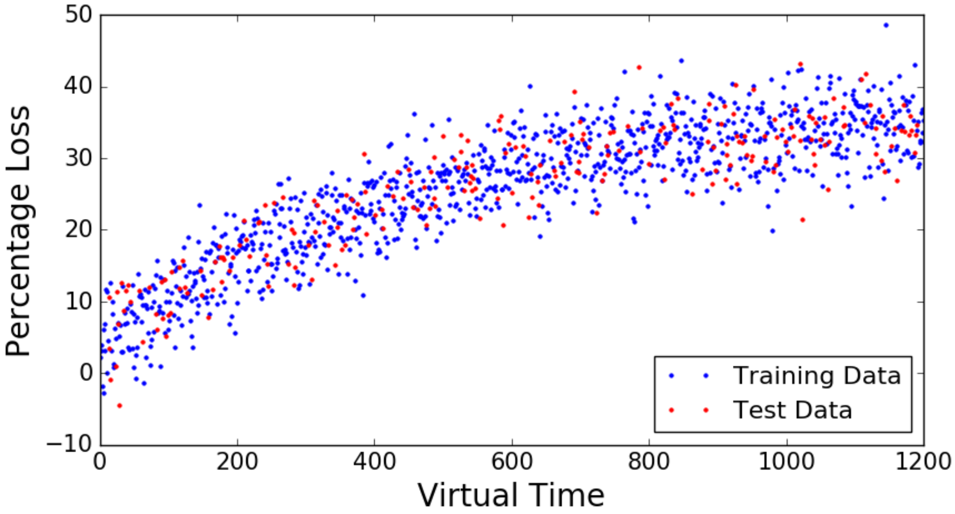


Figure 5.5: Data generated from non-linear model with added noise, divided into a training and a test set.

Using this non-linear model, we assume that time is the only parameter that affects the speed loss. There are no events that affect the performance loss such as damaging of the hull or propeller polishing.

5.1.2.1 Least Squares Regression

We now want to use polynomial least squares regression in combination with cross-validation to find an optimal degree of polynomials to include in the regression. As the model used to generate the data is non-linear, it is expected that some higher degree polynomial will give the best estimated test error. We use polynomials from degree 0 to 7 such that the regression model looks like

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{a} \mathbf{x}_j \beta_j, \tag{5.5}$$

where $a = 0, 1, \cdots, 7$ and $\mathbf{x}_j = \mathbf{x}^j$. We will then perform a 5-fold cross validation for each of the 8 cases to find the optimal degree for α. The cross-validation process can be seen in the top of Figure 5.6, where a polynomial of degree three gives best estimated test error.

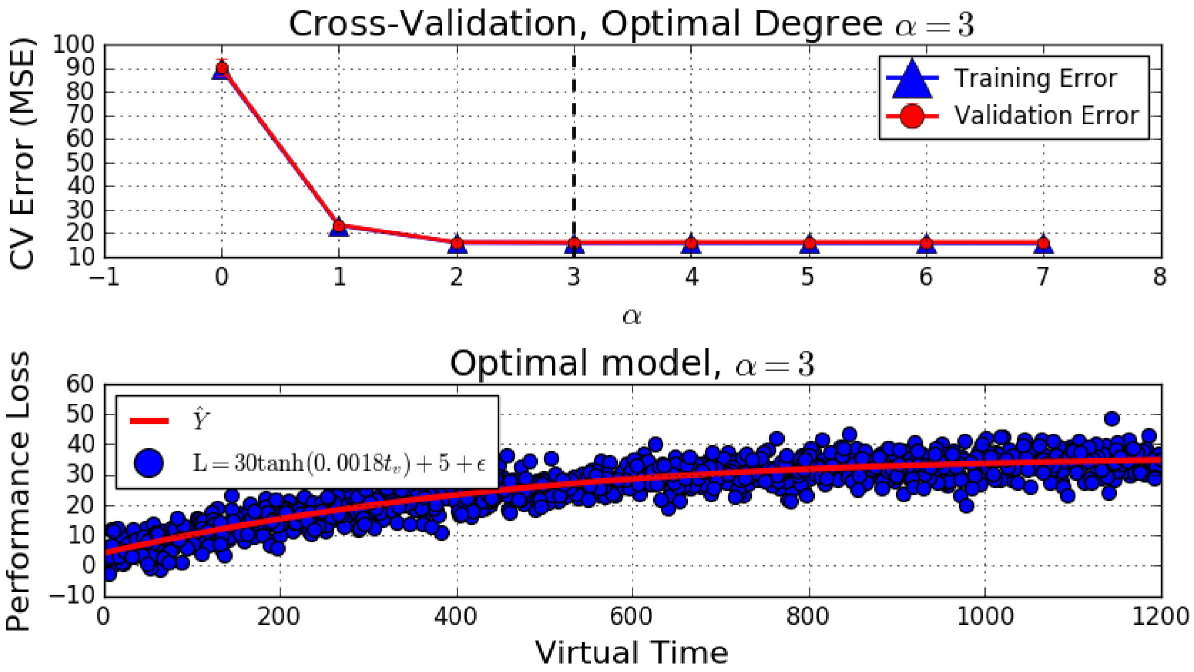


Figure 5.6: Top: Cross Validation to find optimal polynomial degree. Vertical black-dotted line represents the optiaml degree. Bottom: Polynomial regression line for optimal degree.

In the bottom of Figure 5.6 the regression line for $\alpha = 3$ is plotted together with the data. In Table 5.2 we see that the error for $\alpha = 2, \ldots, 7$ does not differ much and they all have relatively low standard deviations. Since the error when $\alpha = 2$ is within one standard deviation from the optimal degree $\alpha = 3$ we would choose a polynomial of degree 2 since this makes the regression model easier to interpret. By choosing $\alpha = 2$ the test error on the independent test set becomes $\mathrm{Err}_{\mathcal{T}} = 14.89$, and the base error is reduced by 83.5 %.

α	**Estimated Test Error** (Err)	**Standard Deviation**
0	90.34	3.83
1	23.44	0.20
2	16.03	0.14
3	15.91	0.12
4	16.01	0.12
5	16.02	0.11
6	16.01	0.11
7	15.99	0.12

Table 5.2: Estimated test error and standard deviation for polynomial regression on non-linear model for different α

5.1.2.2 Nearest-Neighbor Regression

We now use the uniform nearest-neighbor regression. To find the optimal number of neighbors we use a 5-fold cross-validation. This process is seen in the top of Figure 5.7. We have tested for neighbors $k = 1, 2, \ldots, 70$. Since this model is slightly more complicated than the linear model in Section 5.1.1 we expect a fewer number of neighbors to be chosen for the optimal model.

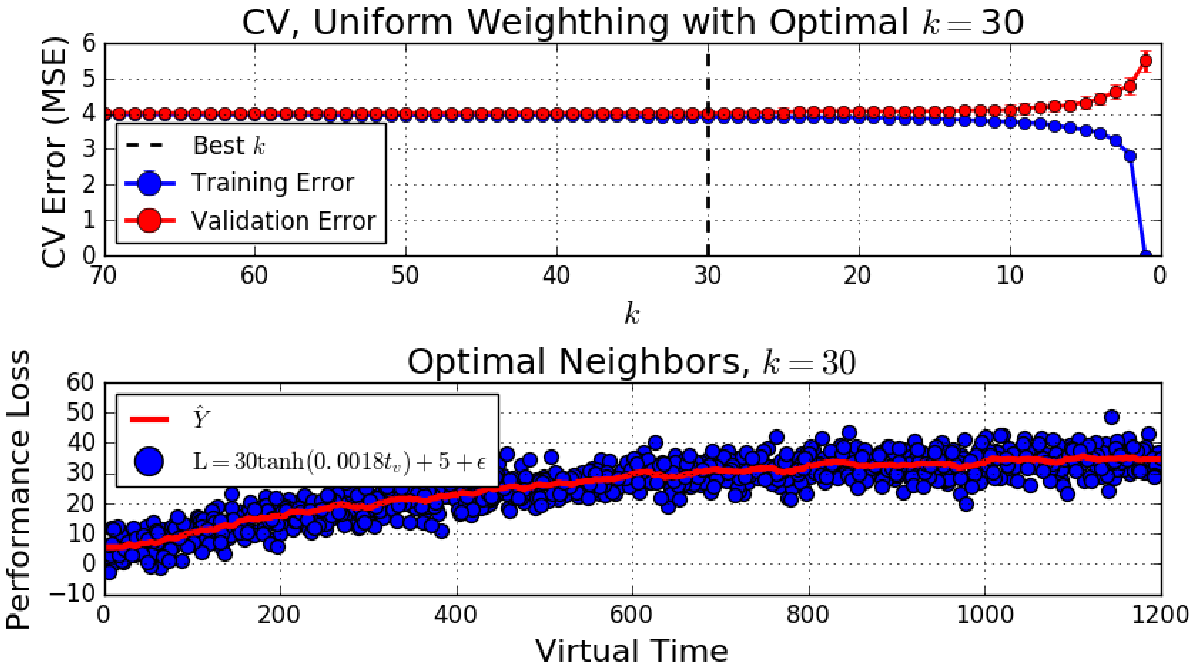


Figure 5.7: Top: Cross Validation to find optimal number of neighbors. Vertical black-dotted line represents the optimal k. Bottom: nearest-neighbor regression line for optimal k.

From the figure we see that the optimal number of neighbors is 30 which is less than for the linear model, as expected. In the bottom of Figure 5.7 the regression line for $k = 30$ is plotted together with the data. For $k = 30$ the test error is $\text{Err}_{\mathcal{T}} = 14.71$. This means that the base error is reduced by 83.7 %, slightly better than a second degree polynomial.

5.1.3 Multidimensional Non-Linear Model

In this case we assume that the speed loss of the vessel is a non-linear function of virtual time t_v and that virtual time is a function of speed over ground (sog), sea water temperature (sw) and real time t_r. That is

$$\text{L} = f(t_v) + \epsilon = f(t_r, \text{sw}, \text{sog}) + \epsilon \tag{5.6}$$

where $\epsilon \sim N(0,4)$. By doing this we assume that the speed of the vessel (sog) and sea water temperature (sw) affects how much loss is added each day, i.e. affect the virtual time such that one virtual day is shorter or longer than one real day. For example, if the vessel is sailing in cold water, it is expected that the fouling is not as rapid as it would be in warmer waters and one virtual day become less then a real day.

How it is assumed sog and sw affect the virtual time can be seen in Figure 5.8. From the figure we see that a temperature of 25°C would speed up the fouling process by two days, such that one real day becomes three virtual days. The sog of the vessel can also be seen in this figure, and can increase or decrease the virtual time by maximum ±0.5 days. If we follow this model, a vessel sailing constantly at 10 [kts] in 5°C sea water, we would have that $t_r = t_v$.

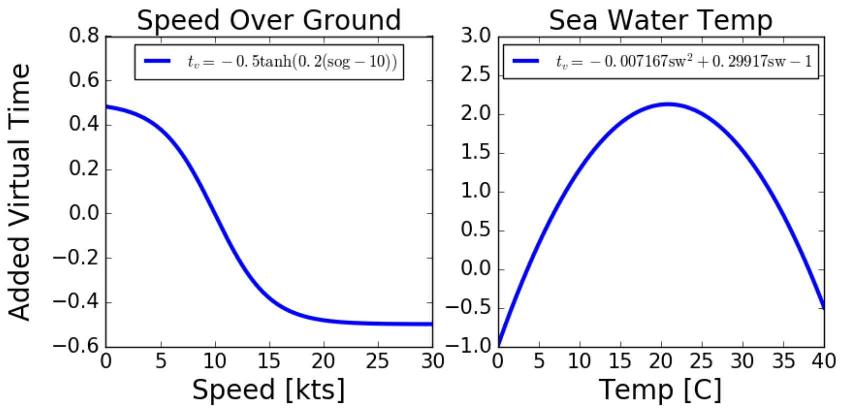


Figure 5.8: Model of how the sea water temperature and speed over ground affect the virtual time.

The full function for the virtual time at day i is then given by

$$t_{v,i} = t_{v,i-1} + t_r + A_{\text{sw}}\text{sw}^2 + B_{\text{sw}}\text{sw} - C_{\text{sw}} - A_{\text{sog}}\tanh(B_{\text{sog}}(\text{sog} - C_{\text{sw}})) \tag{5.7}$$

where $t_r = 1$ (one day increments) and the parameters A, B and C are chosen as seen in Table 5.3.

Parameter	**Value**	**Parameter**	**Value**
A_{sog}	-0.5	A_{sog}	-0.007167
B_{sog}	0.2	B_{sog}	0.29917
C_{sog}	10	C_{sog}	1

Table 5.3: Parameters chosen for the multidimensional non-linear model

To generate sea water temperatures for 1200 days, we divide the into three periods of 400 days each. For each period a random walk (Pearson, 1905) will be started at different starting points. This is done to keep the continuity of real sea water temperature. The random walk used to generate sea water temperature timeseries can be seen at the top of Figure 5.9. We use the normalized cumulative density function of the real speed over ground to generate data similar to that of the real vessel. The generated speed over ground timeseries can be seen at the bottom of Figure 5.9. This generated data actually contains negative sea water temperatures, which is unlikely in the real world.

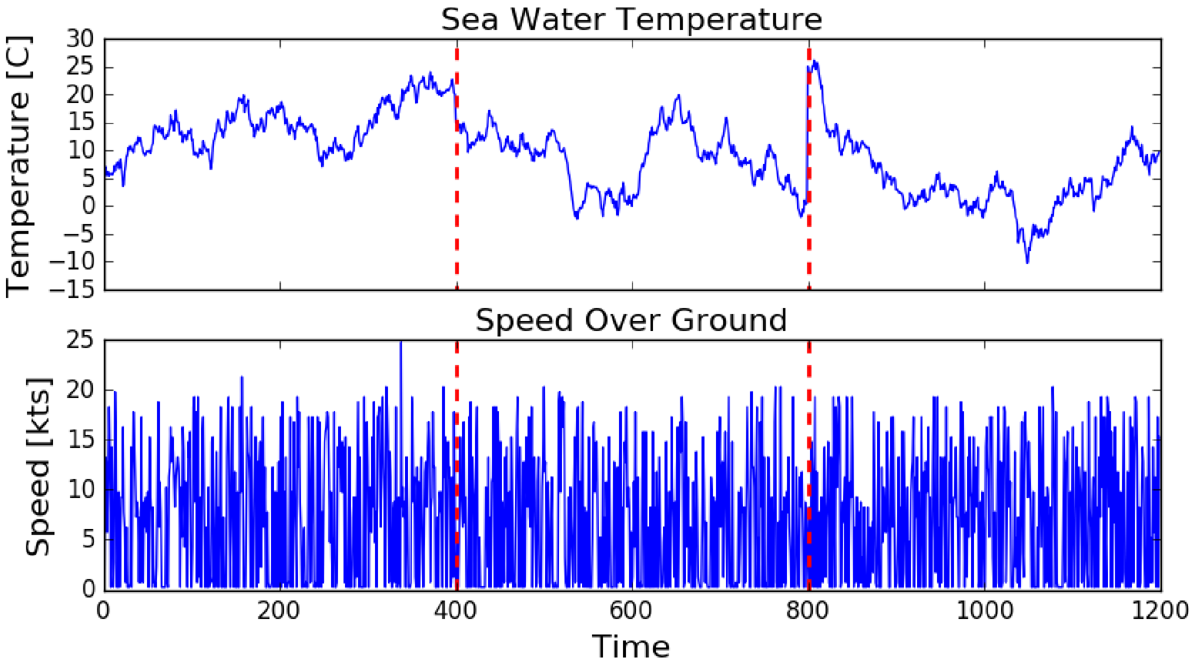


Figure 5.9: Generated sea water temperature from a random walk and generated speed over ground from real data speed distribution. The red dotted lines divides the data into three equally sized periods.

We further assume that the vessel has self-polishing paint, such that the performance loss is slightly decreasing the first 200 days. This is done by having a different function when $t_r < 200$.

We generate data from Equation (5.8) and divide them randomly into a test set of 20 % and a training set of 80 %. In the top of Figure 5.10 the generated data can be seen before noise is added. In the bottom of the figure the data with added noise and divided into a training and a test set can be seen. We notice the clustering of points around $t_v = 1500$ and $t_v = 2200$, this is effects of the virtual time being a function of several variables.

$$L = \begin{cases} 15\tanh(0.0015t_v - 2) + 13.2, & \text{if } t_r < 200 \\ -2.5\tanh(0.025t_v - 2) + 2.5, & \text{otherwise} \end{cases} \tag{5.8}$$

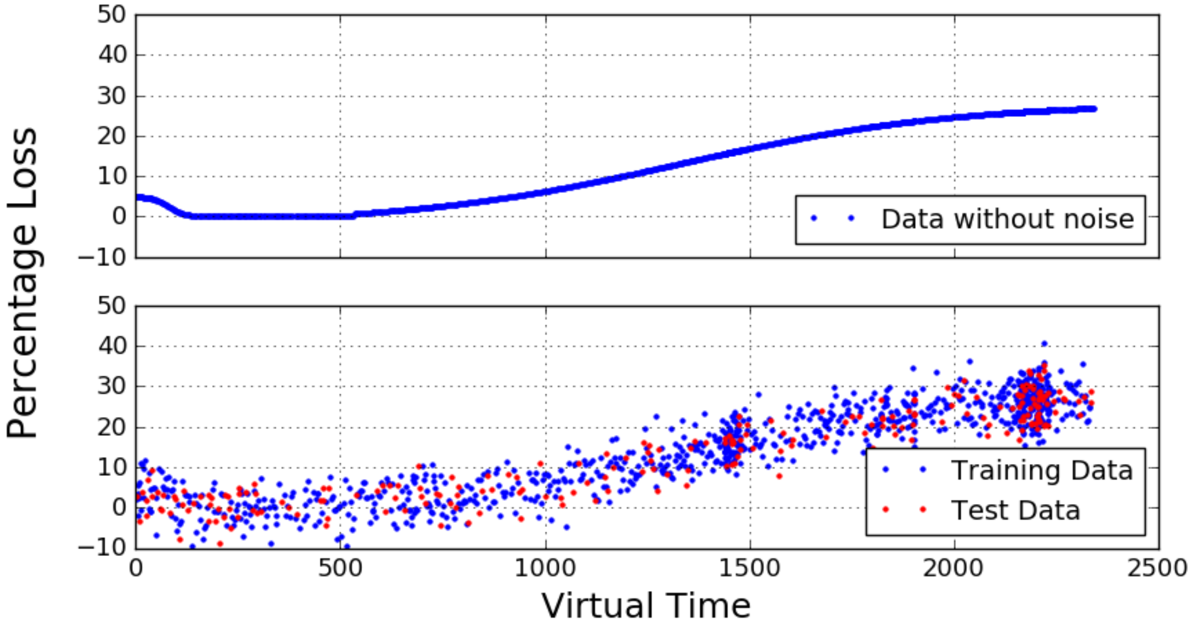


Figure 5.10: Performance loss without (top) and with noise (bottom) for multidimensional non-linear model. When the noise is added the data is divided into a training and a test set.

In this model, it is assumed that the speed over ground and sea water temperatures are mean values for each day, as the time is increased by 1 day each step. In Figure 5.10 we can see that for our model because the vessel is often sailing at low speed and at relatively warm water, approximately 2300 virtual days has passed when only 1200 real time days have passed. In other words, the fouling process of the vessel has doubled compared to a vessel sailing constantly at 10 [kts] and only sailing in waters of 5°C.

5.1.3.1 Linear Least-Squares Regression

We start by using linear least-squares regression on the multidimensional non-linear model we have created. There are three input variables, t_r, sog and sw. The target value is the performance loss L. As the data is generated from a non-linear model with three input variables the linear regression is not expected to get a good test error, but we still expect some reduction of the base error. By using linear least-squares we will find a regression function on the form

$$L = \beta_0 + \beta_t t_r + \beta_{\text{sw}} + \beta_{\text{sog}} \tag{5.9}$$

As this is a multidimensional function with three inputs, it is not easy to visualize the regression plane in a good way. Due to this, we plot the residuals versus fitted values and the distribution of the residuals. This way we get an impression on how well the regression line is predicting new values and if there are any pattern in the residuals. To the left in Figure 5.11, the residuals are plotted against the fitted values from the training set. We see that the residuals appear to follow an s-shaped line. This is an indication that the non-linear relationship between the input variables and the target values was not explained by the regression model.

To the right in Figure 5.11, the standardized residuals are plotted against the theoretical quantiles. This plot gives us information on whether the residuals are normally distributed. As we see the R^2-value is high and the plots follow a straight line, indicating that the residuals are quite normally distributed. Keep in mind that this just a visual check, not an air-tight proof, so it is somewhat subjective.

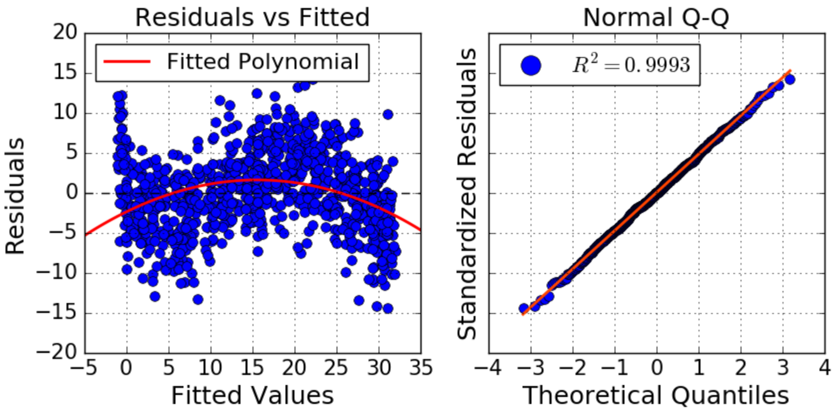


Figure 5.11: Left: Residuals plotted against the fitted values from the training set. Right: Standardized residuals plotted against the theoretical quantiles.

Using this linear regression line, the β's in Equation (5.9) becomes $\beta_0 = -0.853$, $\beta_t = 0.0278$, $\beta_{\text{sw}} = -0.0763$ and $\beta_{\text{sog}} = 0.0251$. The test error $\text{Err}_{\mathcal{T}} = 19.48$ and the base error is 117.68, which gives a base error reduction of 83.4 %, not bad for a simple linear regression.

5.1.3.2 Polynomial Regression

We want to use polynomial least squares regression in combination with cross-validation to find an optimal degree of polynomials to include in the regression. It is expected that a high degree of polynomials will give the lowest estimated test error. However, if a lower degree polynomial only has a slightly higher expected error, the lower degree polynomial will be favored for easier interpretation. It is also expected that t_r will be the most important variable and will hence have higher coefficients then the other variables. We use polynomials from degree 0 to 7 such that the regression model looks like

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{a} \mathbf{x}_j \beta_j, \tag{5.10}$$

where $a = 0, 1, \cdots, 7$ and $\mathbf{x}_j = \mathbf{x}^j$. We will then perform a 5-fold cross validation for each of the 8 cases to find the optimal degree for α. The cross-validation process can be seen in Figure 5.6, where a polynomial of degree 4 gives best estimated test error.

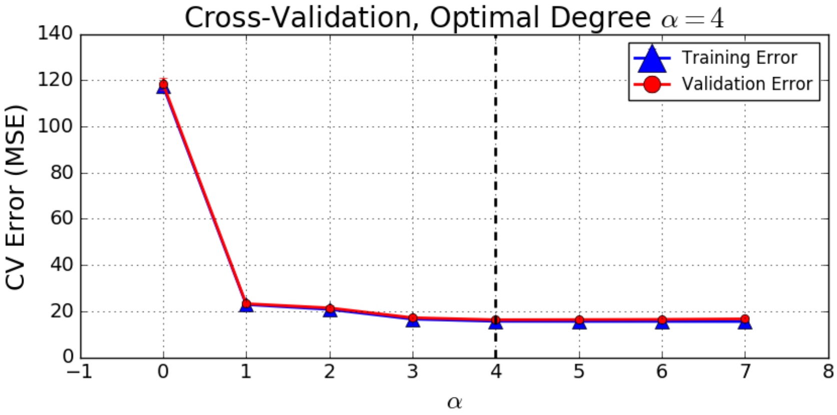


Figure 5.12: Cross-validation for polynomial regression on multidimensional non-linear model. Vertical black-dotted line represents the optimal polynomial degree.

α	**Estimated Test Error** (Err)	**Standard Deviation**
0	118.48	2.59
1	23.22	1.22
2	21.40	1.23
3	17.15	0.54
4	16.25	0.50
5	16.32	0.50
6	16.42	0.50
7	16.63	0.52

Table 5.4: Estimated test error and standard deviation for polynomial regression on multidimensional non-linear model for different α

The expected test errors can be seen in Table 5.4 together with their standard deviation. For a polynomial of degree 4, we have that $\text{Err}_{\mathcal{T}} = 15.34$. This is an expected reduction of 86.9 % compared to the base error.

As we did for the linear regression model, we investigate how the residuals are behaving. To the left in Figure 5.13, we can no longer see a clear pattern for the residuals as we did in the linear regression model. This suggests that the non-linear relationship between input variables and target values are somewhat described by our regression model. However, we see many points for low and high values on the fitted values-axis. This means that the model is struggling to fit high and low values for the performance loss L. This suggests that the assumption of constant variance may not hold. The normal Q-Q plot gives a high R^2-value, suggesting that the residuals are normally distributed.

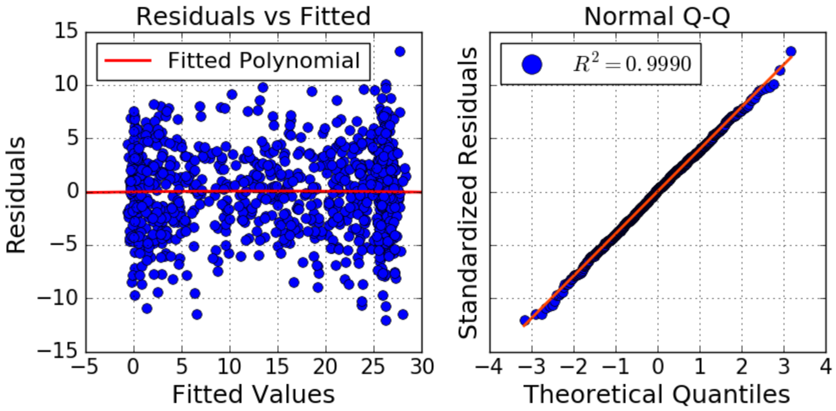


Figure 5.13: Left: Residuals plotted against the fitted values from the training set. Right: Standardized residuals plotted against the theoretical quantiles.

We want to investigate which variables describes the most variance in the fourth-degree regression polynomial. To do this we test all subsets of variables and choose the subset of size k that gives the lowest expected test error by the use of cross-validation. In Figure 5.14 the residual sum of squares and the expected test error for a 5-fold cross-validation is plotted. Each RSS and MSE is plotted for the best subset of the given subset size k.

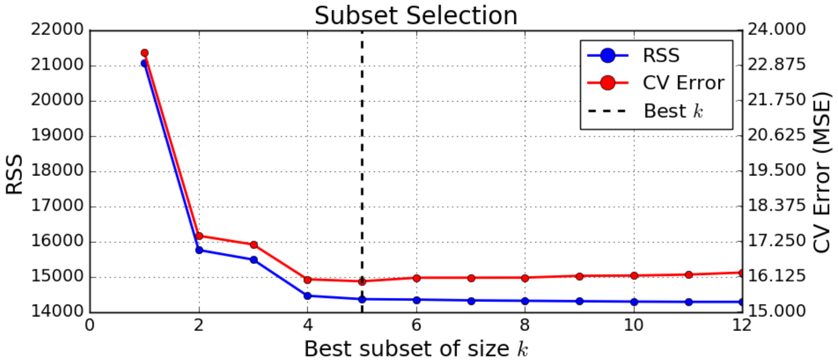


Figure 5.14: Cross-validation used to find the best subset of variables to include in our fourth degree regression polynomial.

We see from the figure that a subset of size $k = 5$ gives the lowest Err. In Table 5.5 the RSS, MSE (Err) and the chosen variables for each best subset of size $k = 1, 2, \ldots, 5$ can be seen. We see that t_r is clearly the most represented variable in the model as expected. By using the 5 optimal variables we can create a regression model which gives a test error of 15.52, which reduces the base error by 86.8 %. This is very close to when all variables were in the model, suggesting that sog and sw are almost insignificant in this linear regression model or that this linear regression model is not able to capture the relation between performance loss, sw and sog. We will probably see similar problems in the real-world data.

Subset size k	**RSS**	**MSE**	**Variables**
1	21065	23.28	t_r
2	15753	17.43	t_r^2 t_r^3
3	15482	17.15	t_r t_r^2 t_r^3
4	14455	16.04	t_r t_r^2 t_r^3 t_r^4
5	14359	15.97	t_r t_r^2 t_r^3 t_r^4 sog^4

Table 5.5: Subset selection table for multidimensional non-linear model for $k = 1, 2, \ldots, 5$.

From the table, we notice that the sub-optimal stepwise subset selection methods would give a wrong subset for $k = 2$. The stepwise method would first find t_r then add the variable that decreases the CV error the most of the remaining variables. For $k = 2$, t_r is not included in the optimal subset. The regression coefficients from the polynomial regression model with 5 optimal variables can be seen in Appendix E.1.

5.1.3.3 Nearest-Neighbor Regression

Finally, we apply the uniform nearest-neighbor regression to our multidimensional non-linear model. To find the optimal number of neighbors we use a 5-fold cross-validation. This process is seen in Figure 5.15. We have tested for neighbors $k = 1, 2, \cdots, 50$. The number of neighbors is expected to be relatively low as the model is quite complex.

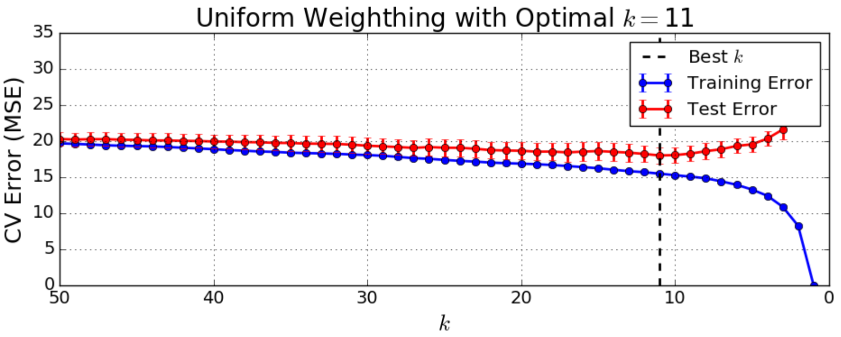


Figure 5.15: Cross-validation for nearest-neighbor regression on multidimensional non-linear model. Vertical black-dotted line represents the optimal number of neighbors k

From the Figure we see that the optimal number of neighbors is quite low as expected $k = 11$. This gives a test error $\mathrm{Err}_{\mathcal{T}} = 16.52$, slightly worse than the best least-squares regression model. Even if the nearest-neighbor regression models gives a good test error, it is hard to understand the model and assess the importance of various variables.

5.2 Real-World Data

In Figure 5.16 the performance loss from the real-world prepared dataset is plotted. The mean loss for each month can be seen in the box to the right of the figure. The grey boxes in the figure indicate the mean ±1 standard deviation. The performance loss is calculated as a percentage loss compared to the expected performance calculated from CFD as discussed in Section 2.2. Again the data is split randomly into a training set of 80 % and a test set of 20 %. There is not a clear pattern of the performance loss like we saw in the simulated data. Hopefully, our regression models will still be able to make models that can predict our test set with low error. Since this is the real-world data, a more in-depth analysis will be carried out then for the simulated data.

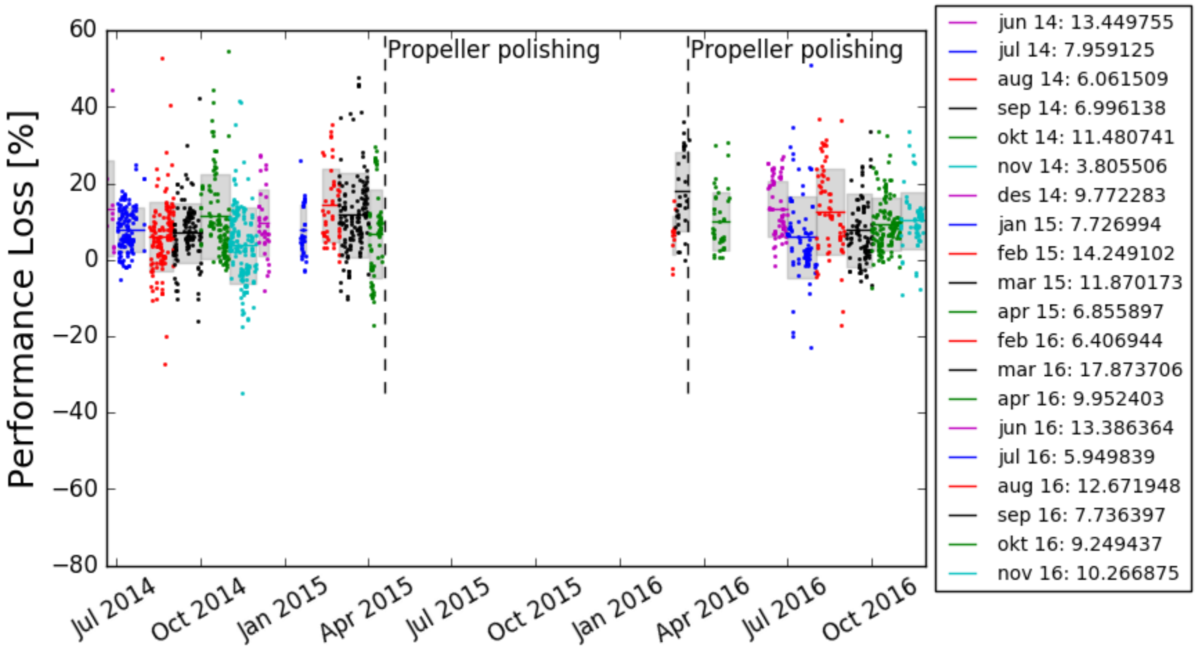


Figure 5.16: The performance loss in the real-world data for the available period. Right box indicates the mean for each month of sailing. Gray boxes in the plot indicates mean ±1 standard deviation.

5.2.1 Least Squares Regression

We start out by taking all variables to the power of $1,2,\ldots,7$ such that we have a maximum of $12 \times 7 = 84$ variables. To choose which degree of polynomial to include in our regression model, we use cross-validation and choose the degree that gives the lowest expected test error Err as seen in Figure 5.17.

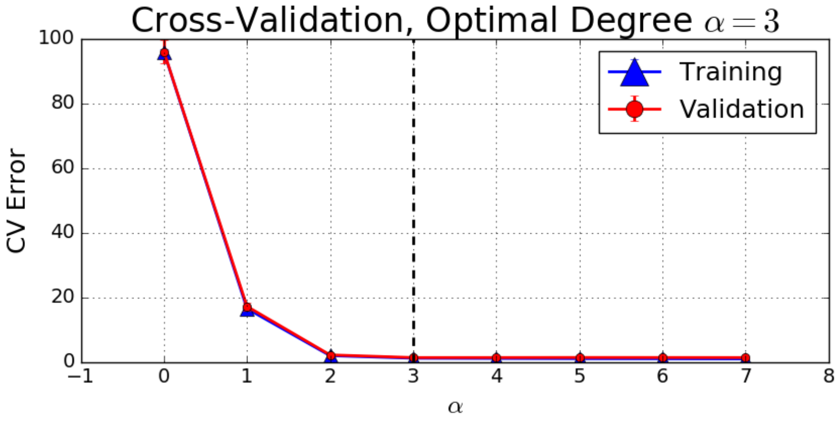


Figure 5.17: Cross validation to choose optimal degree of polynomial regression.

We see that we get the lowest CV error for $\alpha = 3$, but already when $\alpha = 2$ the CV error is quite low. The base error for the real data is 95.85. In Table 5.6, the estimated test errors, as well as standard deviations can be seen. For further analysis, we choose a polynomial of degree 2 for easier interpretation. A polynomial of degree two is able to reduce the error by 97.8 % compared to the base error. This is a significant reduction in error, and better than expected.

α	**Estimated Test Error** (Err)	**Standard Deviation**
0	95.93	3.64
1	17.15	1.21
2	2.14	0.18
3	1.31	0.16
4	1.32	0.16
5	1.34	0.16
6	1.33	0.15
7	1.32	0.15

Table 5.6: Estimated test errors for polynomial regression on real-world data

When choosing a seconds degree polynomial we have a regression model with 24 variables and coefficients, 25 coefficients if we count the intercept β_0. These coefficients can be seen in Table E.2 in Appendix E.1.

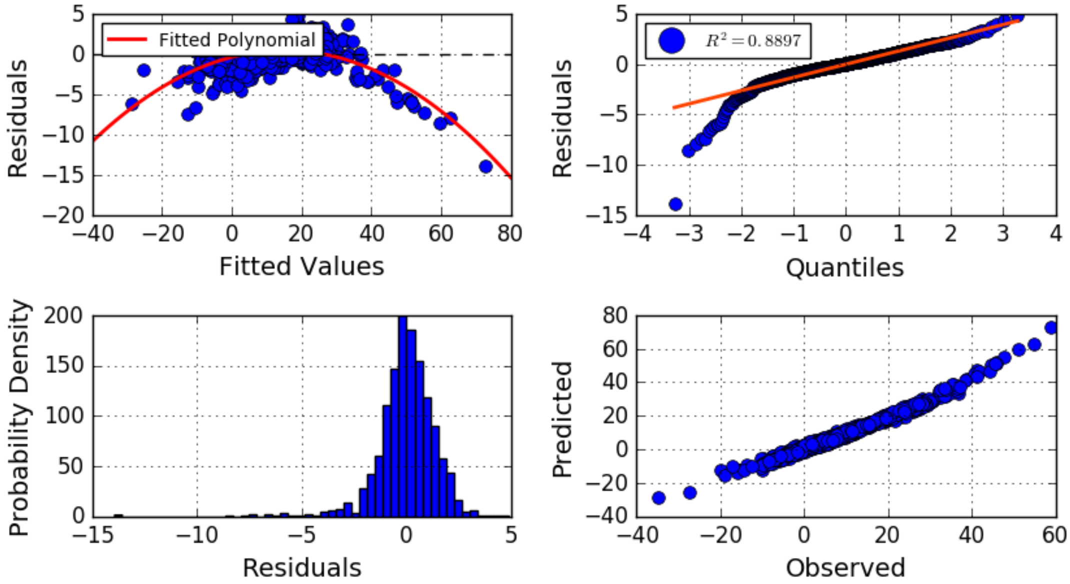


Figure 5.18: Top left: Residuals plotted against the fitted values from the training set. Top right: Standardized residuals plotted against the theoretical quantiles. Bottom left: Probability density of residuals. Bottom right: Observed versus fitted values.

In Figure 5.18 four different plots that give information about the regression model is plotted. To the top left, the residuals are plotted against the fitted values. There is a clear parabolic shape in this shape, which tells us that there are non-linear relationships between input variables and target values that are not described fully by our regression model. This is verified by the Q-Q plot to the top right where we see that the residuals are not normally distributed and seems to be negative or left-skewed. In the bottom left of the Figure the probability density of the residuals is plotted. We can confirm that the residuals are slightly left-skewed. To the bottom right, the predicted versus observed values can be seen. They seem to follow quite a straight line, indicating a clear relation between predicted and observed values.

As we are interested in how time affects the performance loss we make a regression model using only *days since drydock* and *days since propeller polish* as input variables. We add these two variables to the power of $\alpha = 0, 1, 2, \ldots, 7$ to see if the relation between the performance loss and time might be of higher order. We use 5-fold cross-validation to find the degree of input variables that gives the lowest estimated test error. This can be seen in Figure 5.19. A polynomial of degree 6 gives the lowest expected test error, but we can see that for all α, the standard deviations are large, as indicated by the vertical lines. We see that taking the mean of the input variables ($\alpha = 0$) is very close to being within one standard deviation from the optimal degree ($\alpha = 6$). The estimated test error for $\alpha = 6$ is 92.04 ± 3.68, which only gives a base error reduction of about 4 %.

For $\alpha = 1$ we get the coefficients (β) related to *days since drydock* and *days since propeller polish* as 0.734 and 0.598, suggesting that there is a slight increase in performance loss as the time increases. For higher degree polynomials it is more complicated to see the impact of the input variables as one variable squared might cancel out the same variable cubed.

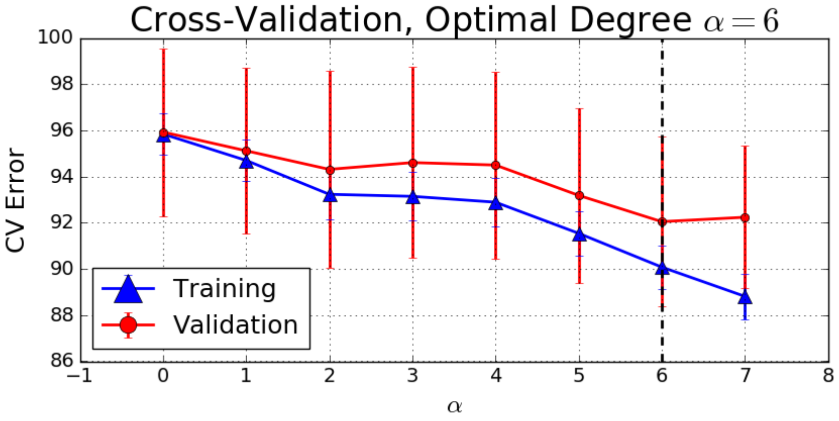


Figure 5.19: Cross validation to choose optimal degree of polynomial regression when the time is the only variables in the input data.

In Figure 5.20 four different plots that give information about the regression model can be seen. We can see that the residuals seem to be normally distributed. In the residuals versus fitted values plot to the top left, we see some values far away from the large cluster of points. This indicates that some outliers have not been removed during the preprocessing of the data. In the Q-Q plot to the top right, we see that the residuals start to swing off at the high and low quantiles, suggesting that the tails of our distribution are not normally distributed.

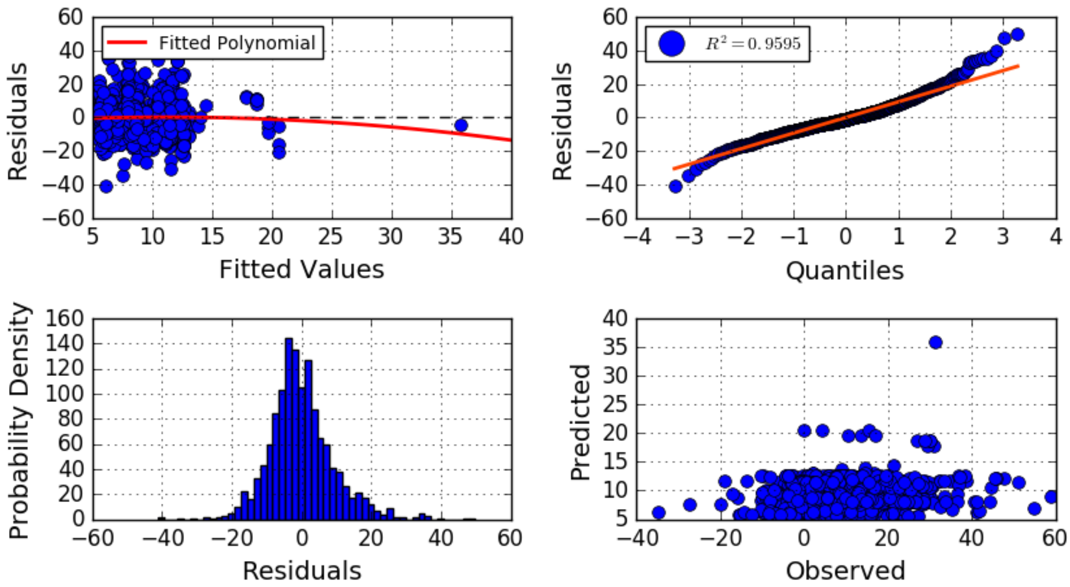


Figure 5.20: Top left: Residuals plotted against the fitted values from the training set. Top right: Standardized residuals plotted against the theoretical quantiles. Bottom left: Probability density of residuals. Bottom right: Observed versus fitted values.

By investigating the relation between the time and the performance loss it is clear that there is no significant relation that can be found using linear regression models. The small relation between the time and the performance loss could as well be random or due to the uncertainties and assumptions made.

5.2.2 Variable Assessment

We want to investigate which variables describes the most variance in our data. We will first use the best subset selection on the 12 variables. Then we will square every variable and find the best subset when we have 24 variables. To do this we test all subsets of variables and choose the subset of size k that gives the lowest expected test error by the use of cross-validation. Ideally, we should run polynomial cross-validation in loop with best subset selection, such that all possible combinations for both polynomial degree and subset of variables are tested together. This is computationally costly, even for as few as 24 variables.

First, we use the 12 chosen variables in the prepared dataset. In Figure 5.22, the residual sum of squares and the expected test error for a 5-fold cross-validation is plotted. RSS and MSE (Err) are plotted for the best subset of the given subset size k.

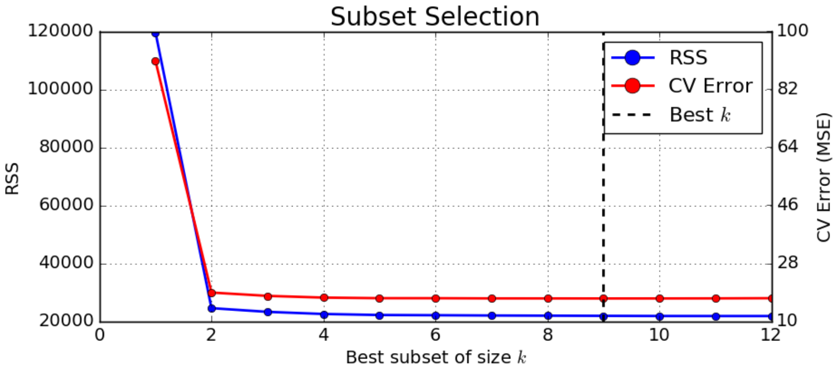


Figure 5.21: Cross-validation used to find the best subset of variables to include from our 12 variables.

From the Figure, we see that a subset of 9 variables gives the lowest CV error. The RSS, CV error and the optimal variables for each subset of size $k = 1, 2, \ldots, 12$ can be seen in Table 5.7. The best subset, $k = 9$, gives a CV error of 17.06 which decreases the base error by 82.2 %. We notice that a stepwise subset selection would produce the same results as we got here since one variable is added for each k. We also notice that the CV error drops slowly after $k = 2$ suggesting that *speed over ground* and *total shaft torque* is the two clearly most important variables in this linear regression model. For $k \geq 4$, *days since drydock* is chosen as a part of the best subset of variables. For $k = 4$ the coefficients (β) related to *speed over ground* and *total shaft torque* calculated by the least-squares regression is -25.3 and 34.5 respectively. The coefficient related to *days since drydock* is only -0.99. This suggests that the performance loss is decreasing slightly as the days increase, the opposite of what one would expect. All coefficients for the optimal subset for $k = 9$ can be seen in Appendix E.1.

Subset size k	**RSS**	**MSE**	**Variables**
1	119669	90.82	Speed over ground
2	24503	18.90	– Total shaft torque
3	23225	17.87	– Total mg power
4	22490	17.32	– Days since drydock
5	22138	17.15	– Water depth
6	22064	17.14	– Total shaft power
7	21968	17.09	– Average draft
8	21905	17.08	– Total shaft speed
9	21848	17.06	– Atmospheric temp
10	21796	17.07	– Days since propeller polish
11	21782	17.09	– Sea water temp
12	21779	17.15	– Speed through water

Table 5.7: Subset selection for the real-world data for $k = 1, 2, \ldots, 12$. '–' means the variables for the previous k.

We now square all our 12 variables and add them to the input variables, such that we have 24 variables. Again we use best subset selection in combination with cross-validation to find the optimal subset of variables. However, this is only done for $k = 1, 2, \ldots, 8$ due to computational cost. The subset selection process can be seen in Figure 5.22.

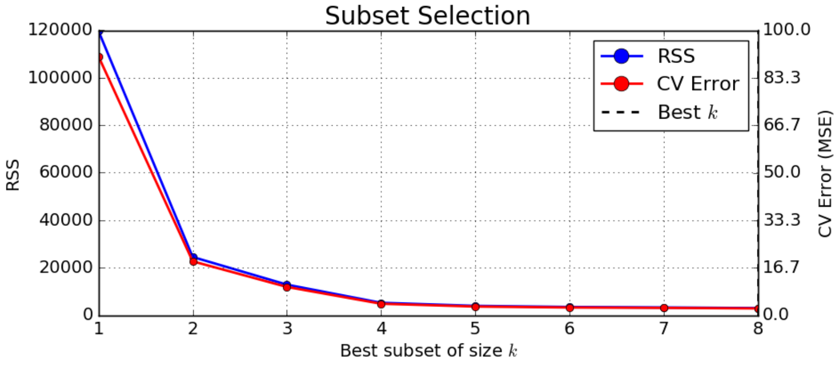


Figure 5.22: Cross-validation used to find the best subset of variables to include in our second degree regression polynomial.

From the Figure, we see that the CV error still decreases when the subset size is $k = 8$. This suggests that we might get lower CV error if we use a larger subset. The computational cost of increasing the subset size increases exponentially, and we would have to calculate around 17 million combinations of subsets if we were to calculate for k up to 24. Like for our multidimensional non-linear model, the stepwise subset selection methods would not give the optimal subsets for some of the k's as some variables that were in the lower k subsets are not in the higher ones. In Table 5.8 the RSS, MSE and the chosen variables for each best subset of size $k = 1, 2, \dots, 8$ can be seen. We see that the only variables represented are the propulsion variables and draft of the vessel. With the 8 optimal variables, the test error becomes 2.48, which reduces the error by 97.4 %.

Subset size k	**RSS**	**MSE**	**Variables**
1	119669	90.82	Speed over ground
2	24503	18.90	Speed over ground Total shaft torque
3	12851	9.91	Speed over ground Total shaft power Total shaft power squared
4	5204	4.00	Speed over ground Speed over ground squared Total shaft power Total shaft power squared
5	3908	3.02	Speed over ground Speed over ground squared Total shaft power Total shaft power squared Average draft
6	3432	2.68	Speed over ground Speed over ground squared Total shaft power Total shaft power squared Average draft Total shaft torque
7	3238	2.55	Speed over ground Speed over ground squared Total shaft power Total shaft power squared Average draft Total shaft torque Total shaft torque squared
8	2963	2.35	Speed over ground Speed over ground squared Total shaft power Total shaft power squared Average draft Total shaft speed Total shaft speed squared Total shaft torque squared

Table 5.8: Subset selection table for real-world data for $k = 1, 2, \ldots, 8$.

In Appendix E.1 the coefficients from using all 24 variables in the regression model, and the coefficients using the subset of the 8 optimal variables in the regression model is listed.

5.2.3 Nearest Neighbor Regression

We apply the uniform nearest-neighbor regression to the real-world data. To find the optimal number of neighbors we use a 5-fold cross-validation. This process is seen in Figure 5.23. We have tested for neighbors $k = 1, 2, \cdots, 50$. The expected number of neighbors is expected to be relatively low as it is expected that the real-world dynamics are quite complex.

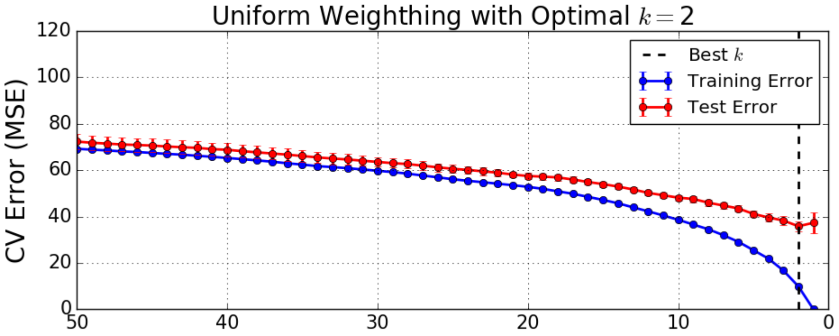


Figure 5.23: Cross-validation for uniformly weighted nearest-neighbor regression on real-world data. Vertical black-dotted line represents the optimal number of neighbors k

From the Figure, we see that the optimal number of neighbors is quite low as expected $k = 2$. This gives a test error 35.45 and reduces the base error by 63.0 %, way worse than for the least-squares regression. We also try the distance weighted nearest-neighbor regression. The cross-validation for this can be seen in Figure 5.24. We the optimal number of neighbors is still $k = 2$ but the test error is slightly better, 32.37 (base error reduction of 66.2 %).

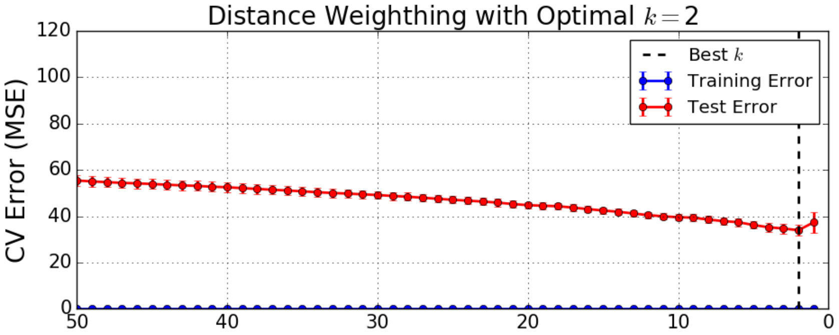


Figure 5.24: Cross-validation for distance weighted nearest-neighbor regression on real-world data. Vertical black-dotted line represents the optimal number of neighbors k

Both nearest-neighbor regressions performed poorly compared to even the simplest least squares regression. This was unexpected, as the nearest-neighbor regression performed equally good as linear regression on our simulated data. As discussed in Section 3.4.2.1, there is no way we can interpret the models made by nearest-neighbor regressions.

Chapter 6

Summary and Recommendations for Further Work

In this Chapter the work done is summarized, and the results are discussed before we provide recommendations for further work. Section 6.1 summarizes the work done, how it was done and whether it was implemented me or not. Section 6.2 presents a brief summary of the work done and the main results. In Section 6.3 the findings are discussed in terms of their strengths and limitations. Finally, Section 6.4 provides recommendations for further work.

6.1 Contributions

All work done in this thesis was done by the use of Python scripts. Originally there were over 3000 CSV-files with more than 1 TB of sensory data stored in an inefficient way. These files were opened and restored as HDF5 files, which reduced the total amount of data to 10.5 GB after the initial variable selection. All handling, conversion, and extraction of data were done by the use of Pandas and NumPy. The find nearest method described in Section 4.4 was implemented by me. The nearest-neighbor outlier removal was done by the help of Scikit-Learn. Scaling (normalization) of the data was done using Scikit-Learn. A self-made implementation of PCA was made to get an understanding of the algorithm, but the implementation from Scikit-Learn was later used due to its simplicity. Scikit-Learn were used for all regression analysis. However, the least-squares regression was implemented independently to get a full understanding of the algorithm. Cross-validation was done by the use of Scikit-Learn. Best subset selection was implemented by me. All plots were made using

Matplotlib. All methods, algorithms and their parameters have been selected by me, and datasets were made by me to get a full understanding of the methods, algorithms and their parameters.

6.2 Summary and Conclusions

Using sensory data from an LNG tanker combined with CFD curves, the relation between vessel performance and various vessel and environmental variables were investigated using a data-driven approach. A total of 12 variables were considered with data for almost three years. The performance loss was calculated by measuring the performance at a given time and comparing it with the expected performance, calculated using CFD. A particular attention was put into investigating the performance loss over time to assess the hull and propeller performance. For analyzing the performance loss, various statistical learning methods were used, such as PCA, linear and non-linear regression.

In Chapter 2 we provided necessary information about the vessel and the available data. The chapter also gave a theory foundation on hull-propeller performance and the background assumptions on which this theory is built. We chose a total of 24 variables from the 338 available variables, based on own experience and discussions with DNV GL and supervisors.

Chapter 3 presented several methods from the field of statistical learning. The theory and interpretation were presented and for some of the methods, a simple example was presented. This chapter also gave a short introduction to preprocessing of data. Outliers and missing values were discussed as well mean centering and scaling of variables. Only a small subset of methods from the broad field of statistical learning were presented. The selected methods were chosen based on practical use, interpretation, and visualization. In the end of this Chapter we discussed methods for model assessment and selection, such that we could measure the performance of various regression models.

In Chapter 4 we preprocessed the given data into a suitable dataset for analysis. All steps and assumptions made along the way were discussed. We ended up with a small number of observations for the final prepared dataset (only 1833 observations), this is a relatively small number of observations and should ideally be much larger. The difference in temporal resolution among the variables forced us to align observations within 20 minutes to one timestamp. By this, we assumed that the state of the vessel and environment does not change within 20 minutes. In the end of this Chapter we did an unsupervised analysis of the prepared dataset. This revealed the variable correlations and which variables described the most of the variance in the dataset. We saw that the variables related to the propulsion system were highly correlated, as we expected. What we also saw was that the *performance loss* and variables related to time described little variance in the first two components. This told us that it might be hard to find a reliable relation between the *performance loss* and time.

In Chapter 5 the methods described in Chapter 3 were used to explore the underlying relationship between the vessel performance and the 12 chosen variables. We first simulated data to mimic the real-world data for three different cases. A methodology for regression was employed to recover the relationships in the simulated data. Here we saw that the regression models were able to uncover the relation between the input variables and the performance loss well. We also saw that the linear regression models performed equally or better than the non-linear nearest neighbor regression. The linear models also allowed us to interpret the results and find the importance of the input variables on the regression model. When investigating the importance of variables for the multidimensional non-linear model, we saw that the speed over ground and sea water temperature were almost neglected in the regression model. This revealed that the linear regression model might struggle to find complex interactions. In the second part of this chapter, we applied the same methodology on the real-world data. The linear regression models gave strong results. A second-degree polynomial reduced the base error by 97.8 %, higher than expected. The non-linear regression models performed much worse, where the best nearest-neighbor regression only reduced the base error by 66.2 %. From the least-squares regression, we saw that the coefficients related to the propulsion system were clearly more important than the rest of the coefficients. When we tried to investigate the relation between time and performance loss, we got insignificant results. Thus, for this data, the time had an insignificant relation to the performance of the vessel and no prognosis model in a context of maintenance operation could be defined. However, the results suggested that the right operational profile can reduce the performance loss significantly.

6.3 Discussion

In this section, we will further discuss some of the results highlighting different limitations and strengths in our methodology.

We have tried to discuss all the known limitations with our methodology in Chapter 3. In the preprocessing of data, we assumed that the parameters of our vessel do not change much within a 20-minute interval. This was done to get our sparse data into a matrix such that we could perform analysis. Due to this assumption, there will be situations where parameters are put on the same timestamp when in reality the operational profile of the vessel has changed, giving a false relationship between variables. We used simple methods for removing outliers, and there are most likely observations in the prepared dataset that should have been removed. However, there is no universal method to remove outliers that work on all datasets, so to limit the scope some methods had to be preferred over others. In general, the removal of data were done carefully due to the relatively small amounts of speed measurements.

Many variables were removed due to strange behavior or because they were thought to be unnecessary. For instance, the rudder angles were removed, they could have been used to indicate drift compensation or turning of the vessel. Due to few measurements and poor resolution of the rudder variables, they were removed. If the speed through water measurements were reliable, this could have improved the results as the speed over ground can be wrong if the speed relative to the water is large as discussed in Section 2.2.

A redefinition of some variables were also done, ignoring variations and differences between subsystems. In this analysis the specific behavior of the different generators, differences between starboard and port shaft etc. were not of interest. This reduced the number of variables and made the interpretation of the results easier. Some effects are lost by doing these redefinitions, for instance, if the vessel is delivering more power to one propeller to compensate for sideways current.

As discussed in Chapter 2 there are many things that can affect the performance loss of a vessel. The data was filtered to find conditions where wind and waves should be of minor importance, but there is no guarantee for this filtering removing all the situations where waves and wind were significant. We are using the wind speed to filter out high waves. There can be high waves even if there is no wind. The ideal would be to have a motion reference uni (MRU) to measure the motions of the vessel. In addition to this, we should have filtered on many more conditions to ensure that the vessel is not turning, in heavy current or that the vessel is not accelerating. More effort should also be put into ensuring relatively same weather conditions.

As mentioned in Chapter 3 there is no statistic or direct measure of success of the results of an unsupervised analysis. The unsupervised methods rely on interpretation, such that with more knowledge and understanding of the system, one could interpret and understand the results better. However, using PCA it was pleasant to find expected correlations. In general, it shows the potential of such methods to provide insight into the data, for example, to get a basic understanding of the system before proceeding with supervised methods. As mentioned in Section 3.3.2 the PCA assumes that the variables are described by Gaussian distribution, which could prove to be an invalid assumption. With this in mind, nonlinear methods like kernel-PCA could be investigated.

The focus of this thesis was more on linear regression methods rather than non-linear. As discussed in Chapter 3 this can make it hard to find any non-linear interactions between variables. However, the linear regression models showed better test errors then the nearest-neighbor regression for the real data by a large margin. The linear methods also made it possible to investigate the importance of the variables to see which variables affect the performance loss. Keep in mind that we did not consider interactions between quantitative inputs and transformation of quantitative inputs. Including more terms like $\mathbf{x}_2 = exp(\mathbf{x}_1)$ or $\mathbf{x}_3 = \mathbf{x}_2 \times \mathbf{x}_1$ could improve the regression models. To find the variable importance of a more complex regression model, a variable sensitivity analysis should be carried out.

As we saw in Chapter 5 the time was an insignificant variable to describe the performance loss for the real-world data. In the multidimensional non-linear model we saw that the speed over ground and sea water temperature were almost non-existent in the linear regression models. It could be the case that the same is happening in the real-world data, where the effects of time are smaller or equal to the noise and being ignored or assigned very small coefficients.

The expected speed for a given power consumption is calculated from CFD curves. These might not be completely accurate. If this is the case then a deviation from the expected speed at a given power consumption would be related to the propulsion. This might be a good reason to why the propulsion variables are overly represented in the regression models compared to the time and environmental variables.

If we fully trust the results found in this regression analysis, the propulsion variables are or much greater importance than time and environmental variables and the time when one wants to reduce the performance loss of a vessel. The data is clearly showing that a combination of the variables and the variables squared are able to explain the performance loss. One reason why we see such a high importance of the propulsion variables could be because the fouling process is slower for a sailing vessel than for a vessel at dock. Another reason the performance loss is not decreasing over the full period might also be due to the hull-paint which is a self-polishing paint delivered by Jotun.

No significant change in performance was seen by the propeller polishing. One good reason for this might be because there is almost no data in between the two propeller polishes. This gives us no data after the first propeller polish and no data before the second propeller polish.

A huge limitation to these results is the lack of data, especially for certain variables like speed. The lack of observations causes the results to be biased. Some of the results might be specific to this relatively short period of three years. The data is also only gathered from one vessel, making it vessel specific. A larger database of several vessels for a longer period would most certainly increase the reliability of the results found in this thesis. In general, the data-driven approach using multivariate analysis and statistical learning proves to be useful when analyzing high-dimensional data with complex variable interactions.

6.4 Recommendations for Further Work

The current work can lead to improvements and exploration in several direction.

Improvement of data quality

When using real-world sensory data an increased amount and/or better quality of data is expected to increase the prediction performance of statistical methods. Subsequently, it will be possible to draw more reliable conclusions. Introducing more data from several vessels, longer periods and more parameters is expected to increase the performance of the methodology in this thesis. Adding more environmental or MRU data could be done to ensure that there is no resistance to overcome wind and waves. This would make the assumption that fouling and calm water resistance is the only forces acting on the vessel stronger. Higher frequency data would make it possible to detect dynamics like turning or accelerating of the vessel.

Methodology improvements and exploration

The detection of positive and negative events, such as a propeller polishing or damaging of the propeller, could be studied as a classification problem. This may give a better way to see the effects of such events as these events vary significantly. The methods applied to the data is also a very small subset of all possible methods available, more sophisticated methods, such as neural-networks, might make a better regression model if tuned properly. One could also use Gaussian process regression, this non-linear method still allow for variable inspection to see the importance of various variables. More effort could be put into analyzing the variables, for instance, by the use of Monte Carlo simulation.

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Bellman, R. E. and Roth, R. S. (1986). *The Bellman Continuum: A Collection of the Works of Richard E. Bellman*. World Scientific.

Bermingham, M. L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., Wright, A. F., Wilson, J. F., Agakov, F., Navarro, P., and others (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5.

Breiman, L. and Spector, P. (1992). Submodel Selection and Evaluation in Regression. The X-Random Case. *International Statistical Review / Revue Internationale de Statistique*, 60(3):291–319.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM.

Briggs, W. (2008). *Example of how easy it is to mislead yourself: stepwise regression.*
http://wmbriggs.com/post/92/
[Accessed: 2017-05-11].

Callow, M. E. and Callow, J. A. (2002). Marine biofouling: a sticky problem. *Biologist*, 49(1):1–5.

Carlton, J. (2011). *Marine propellers and propulsion*. Butterworth-Heinemann.

CERN (2009). *LHC: the guide.* http://cds.cern.ch/record/1092437/files/CERN-Brochure-2008-001-Eng.pdf [Accessed: 2017-05-09].

DataCamp (2017). *Choosing R or Python for data analysis? An infographic.* https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis#gs.tnzXtZc [Accessed: 2017-05-09].

DNVGL (2017). ENVIRONMENTAL CLASS.

Eniram (2012). Study of Hull Fouling on Cruise Vessels Across Various Seas.

Flom, P. L. and Cassell, D. L. (2007). Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use. *NorthEast SAS Users Group (NESUG): Statistics and Data Analysis.*

Fortmann-Roe, S. (2012). *Understanding the Bias-Variance Tradeoff.* http://scott.fortmann-roe.com/docs/BiasVariance.html [Accessed: 2017-05-02].

Gundermann, D. and Dirksen, T. (2016). A Statistical Study of Propulsion Performance of Ships and the Effect of Dry Dockings, Hull Cleanings and Propeller Polishes on Performance.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques.* Elsevier.

Hasselaar, T. W. F. (2011). An investigation into the development of an advanced ship performance monitoring and analysis system.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations.* Springer series in statistics. Springer, New York.

HDFGroup (2017). *Hierarchical Data Format.* https://support.hdfgroup.org/HDF5/whatishdf5.html [Accessed: 2017-05-12].

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Höskuldsson, A. (1994). Data analysis, matrix decompositions, and generalized inverse. *SIAM Journal on Scientific Computing*, 15(2):239–262.

IMO (2009). Second IMO GHG Study.

ISO (2015). ISO/TC 8/SC 2/WG 7, Measurement of changes in hull and propeller performance.

Jotun (2017). SeaQuantum.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Stanford, CA.

Kriegel, H.-P., Zimek, A., and others (2008). Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452. ACM.

Larsen, N. L., Simonsen, C. D., Nielsen, C. K., and Holm, C. R. a. (2012). Understanding the physics of trim. In *Green Ship Technology Conference, Copenhagen*.

Last, M., Kandel, A., and Bunke, H. (2004). *Data mining in time series databases*, volume 57. World scientific.

Little, R. J. and Rubin, D. B. (2014). *Statistical analysis with missing data*. John Wiley & Sons.

Martens, H. and Martens, M. (2001). *Multivariate analysis of quality. An introduction*. IOP Publishing.

Martens, H. and Naes, T. (1992). *Multivariate calibration*. John Wiley & Sons.

Matplotlib (2017). *Matplotlib 1.5.3 Documentation*. http://matplotlib.org/ [Accessed: 2017-05-09].

Mwitondi, K. S. (2013). *Data mining with Rattle and R*. Taylor & Francis.

NOAA (2017). *National Oceanic and Atmospheric Administration*. http://www.noaa.gov/ [Accessed: 2017-05-12].

Pandas (2017). *Pandas - Python Data Analysis Library*. http://pandas.pydata.org/ [Accessed: 2017-05-09].

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572.

Pearson, K. (1905). The problem of the random walk. *Nature*, 72(1865):294.

Pedersen, B. P. (2014). *Data-driven Vessel Performance Monitoring*. DTU Mechanical Engineering.

Perera, L. P. and Mo, B. (2016). Marine Engine Operating Regions under Principal Component Analysis to evaluate Ship Performance and Navigation Behavior. SINTEF.

Python Foundation (1997). *Comparing Python to Other Languages*. https://www.python.org/doc/essays/comparisons/ [Accessed: 2017-05-09].

Python Foundation (2017). *About Python*. https://www.python.org/about/ [Accessed: 2017-05-09].

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438. ACM.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer.

SciKit-Learn (2017). *SciKit-Learn: Machine Learning in Python*. http://scikit-learn.org/stable/ [Accessed: 2017-05-09].

SciPy (2017). *SciPy - Scientific Computing Tools for Python*. https://www.scipy.org/about.html [Accessed: 2017-05-09].

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Spyder (2017). *Spyder IDE Wiki*. https://github.com/spyder-ide/spyder/wiki [Accessed: 2017-12-11].

Tukey, J. W. (1977). *Exploratory Data Analysis*. Pearson, Reading, Mass, 1 edition edition.

US Environmental Protection Agency (2017). *Data on Cars used for Testing Fuel Economy*. https://www.epa.gov/compliance-and-fuel-economy-data/data-cars-used-testing-fuel-economy [Accessed: 2017-05-09].

Vincenty, T. (1975). Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations. *Survey Review*, 23(176):88–93.

Wikipedia (2017). *Beaufort Scale*. https://en.wikipedia.org/wiki/Beaufort_scale [Accessed: 2017-05-12].

Wikipedia (2017). *Moving Average*. https://en.wikipedia.org/wiki/Moving_average [Accessed: 2017-04-09].

Appendix A

Distance Metrics and Proximity Matrix

This appendix provides a mathematical description of the proximity matrix and some of the most frequently used distance metrics in statistical learning.

A.1 Metrics

In mathematics, a metric is a function that defines the distance between two observations $\mathbf{a}$ and $\mathbf{b}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. Table A.1 gives a functional description of some frequently used metrics in machine learning and pattern recognition.

A.2 Proximity Matrix

The data $\mathbf{X} \in \mathbb{R}^{m \times n}$ can be directly represented in terms of the proximity between pairs of observations $(\mathbf{x}_i, \mathbf{x}_k)$. This can either be similarities or dissimilarities. The proximity data can be represented by a matrix $\mathscr{D} \in \mathbb{R}^{m \times m}$, where m is the number of observations.

Names	**Function**
Euclidean distance	$\lVert\mathbf{a}-\mathbf{b}\rVert_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Manhattan distance	$\lVert\mathbf{a}-\mathbf{b}\rVert_1 = \sum_i \lvert a_i - b_i \rvert$
Maximum distance	$\lVert\mathbf{a}-\mathbf{b}\rVert_\infty = \max_i \lvert a_i - b_i \rvert$
Mahalanobis distance	$\sqrt{(\mathbf{a}-\mathbf{b})^T \mathbf{S}^- 1 (\mathbf{a}-\mathbf{b})}$

Table A.1: Frequently used distance metrics.

Most algorithms presume a symmetric proximity matrix of dissimilarities with zero diagonal elements and nonnegative elements CITE

$$\mathscr{D} = \left(d_{ij}\right) \in \mathbb{R}^{m \times m}, \quad d_{ii} = 0, \quad d_{ij} \geq 0, \tag{A.1}$$

where the dissimilarity between two observations $D\left(\mathbf{x}_i, \mathbf{x}_k\right)$ is determined by a weighted combination of the d attribute dissimilarities $d_j\left(x_{ij}, x_{kj}\right)$, $j = 1, 2, \ldots, d$,

$$D\left(\mathbf{x}_i, \mathbf{x}_k\right) = \sum_{j=1}^{n} w_j \cdot d_j\left(x_{ij}, x_{kj}\right). \tag{A.2}$$

When computing the dissimilarity between two observations it is usual to give all attributes equal influence in characterizing dissimilarity. However, as CITEdiscusses, if the goal is to discover natural groupings in the data, variables that affects group separation more should be assigned a higher influence in defining observation dissimilarity. Specifying an appropriate dissimilarity measure is far more important in successful clustering than the choice of clustering algorithms.

Appendix B

Vessel Information

B.1 CFD Curves

Speed-power curves calculated using CFD software calculated by DNV GL. Valid for the case when the vessel is sailing at calm sea (low wind and small waves) and no significant fouling has taken place.

Speed [kts]	**Extended sea trial curve [kW]**
7	895.39
8	1410.40
9	2121.29
10	3014.29
11	4075.64
12	5291.56
13	6648.29
14	8132.05
15.81	11087.64
18.67	15708.78
20.52	20357.13
21.03	22021.57

Table B.1: Expected power consumption for a given speed at 9 m draft calculated by CFD

Speed [kts]	**Extended sea trial curve [kW]**
7	916.295664
8	1528.92761
9	2164.458411
10	2857.679257
11	3643.381339
12	4556.355847
13	5631.39397
14	6903.286899
16.5	11172.57618
17	12229.79967
18	14526.10031
19.77	19667.23658
20.5	22248.83719

Table B.2: Expected power consumption for a given speed at 11.5 m draft calculated by CFD

Appendix C

Preprocessing

In this chapter various information from the preprocessing (Chapter 3) can be seen.

C.1 Simple Filters

Filters used for preprocessing. Table includes the minimum and maximum limit for each variable as well as the number of observations removed for each filter.

Name	Lower limit	Upper limit	Points removed
Cargo Level - Tank 1	0	27	19
Cargo Level - Tank 2	0	27	18
Cargo Level - Tank 3	0	27	20
Cargo Lever - Tank 4	0	27	28
Sea Water Temperature	0	50	10
Atmospheric Temperature	-20	50	2
Speed Over Ground	0	28	15
Speed Through Water	0	28	684
Wind Speed	0	100	15
Wind Relative Direction	0	360	0
Rudder Angle Port	-40	40	0
Rudder Angle Starboard	-40	40	0
Draft Forward	0	50	0
Draft Aft	0	50	0
Main Generator Engine 1 Power	0	12000	84
Main Generator Engine 2 Power	0	12000	111
Main Generator Engine 3 Power	0	12000	324
Main Generator Engine 4 Power	0	12000	36
Shaft Torque Port	-800	2000	8
Shaft Torque Starboard	-800	2000	12
Shaft Speed Port	-50	100	16
Shaft Speed Starboard	-50	100	21
Total Fuel Gas Flow to Main Generators	0	10000	0
Water Depth	-2000	0	99
Heading	0	360	13

Table C.1: Simple min-max filters used on real-world data

C.2 Nearest Neighbor Filter

Nearest-neighbor outlier used to remove 10 % of the observations furthest away from its neighbors can be seen in Figure C.1 (before filtering) and in Figure C.2 (after filtering). The data is normalized before the outlier removal is done. The mean for each month is plotted and the value for each mean can be seen to the right of the figures. The grey boxes in the plots indicates the mean ±1 standard deviation. It is clear that the nearest neighbor outlier removal removes some obvious outliers and also reduces the standard deviations.

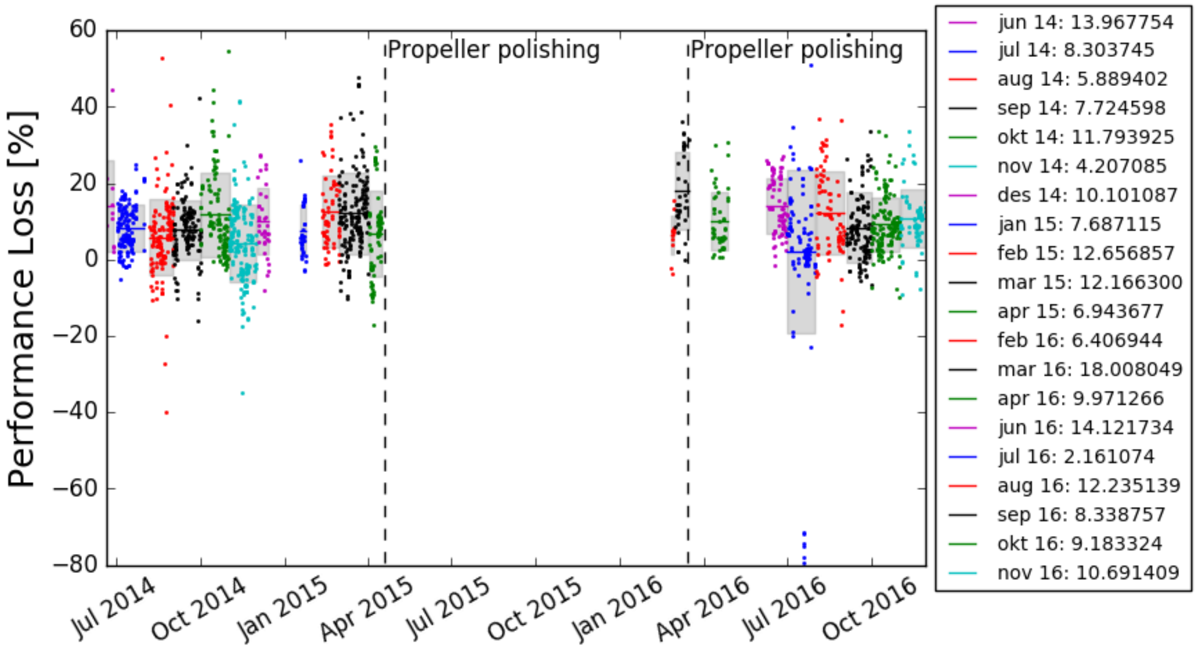


Figure C.1: Unfiltered performance loss.

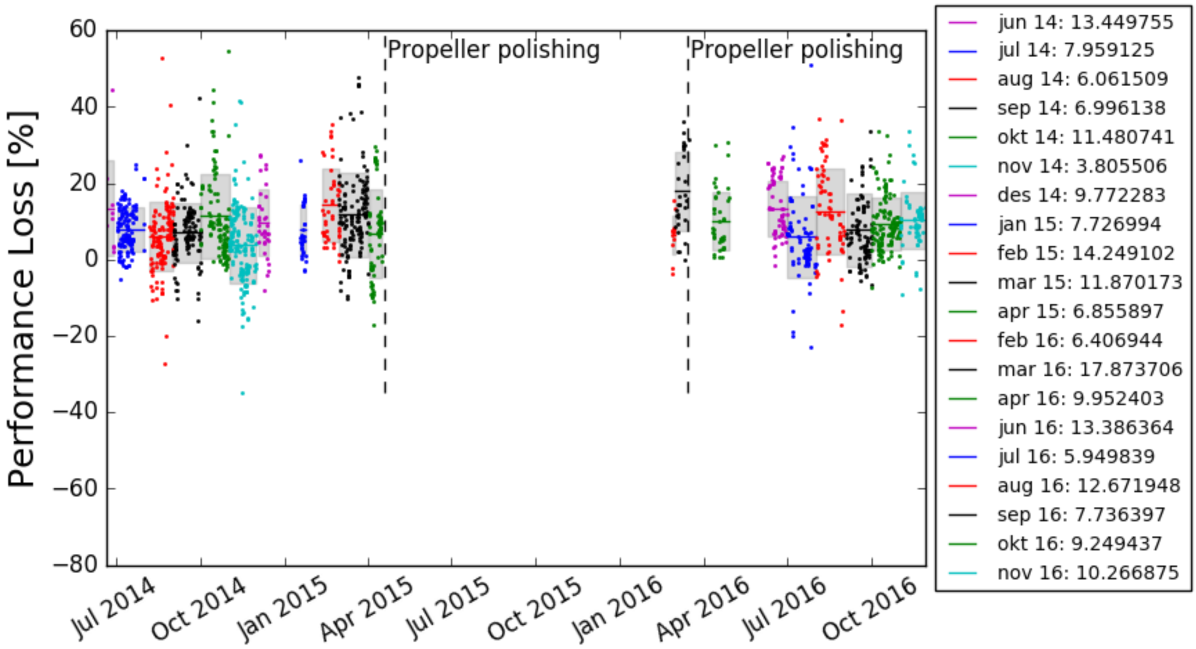


Figure C.2: Performance loss after filtered with nearest-neighbor outlier removal.

Appendix D

Wind

D.1 Calculation of True Wind Speed and Direction

True wind velocity, v_{wt} [m/s], and true wind direction, ψ_{wt} [radians], at height of the anemometer is computed from the relative wind velocity, v_{wr} [m/s], the vessel speed over ground, v_g [m/s], the direction of the relative wind (ψ_{wr} [radians]) and the vessel heading, ψ_0 [radians], according to Formulae (D.1) and (D.2). To the left in Figure D.1 I illustrate the sign convention for directions relative to the vessel heading. To the right in Figure D.1 the true and relative wind speed can be seen, as well as the vessel speed over ground, the true and relative wind direction and heading of the vessel.

$$v_{wt} = \sqrt{v_{wr}^2 + v_g^2 - 2v_{wr}v_g cos(\psi_{wr})} \tag{D.1}$$

$$\psi_{wt} = tan^{-1}\left(\frac{v_{wr}sin(\psi_{wr} + \psi_0) - v_g sin(\psi_0)}{v_{wr}cos(\psi_{wr} + \psi_0) - v_g cos(\psi_0)}\right) \tag{D.2}$$

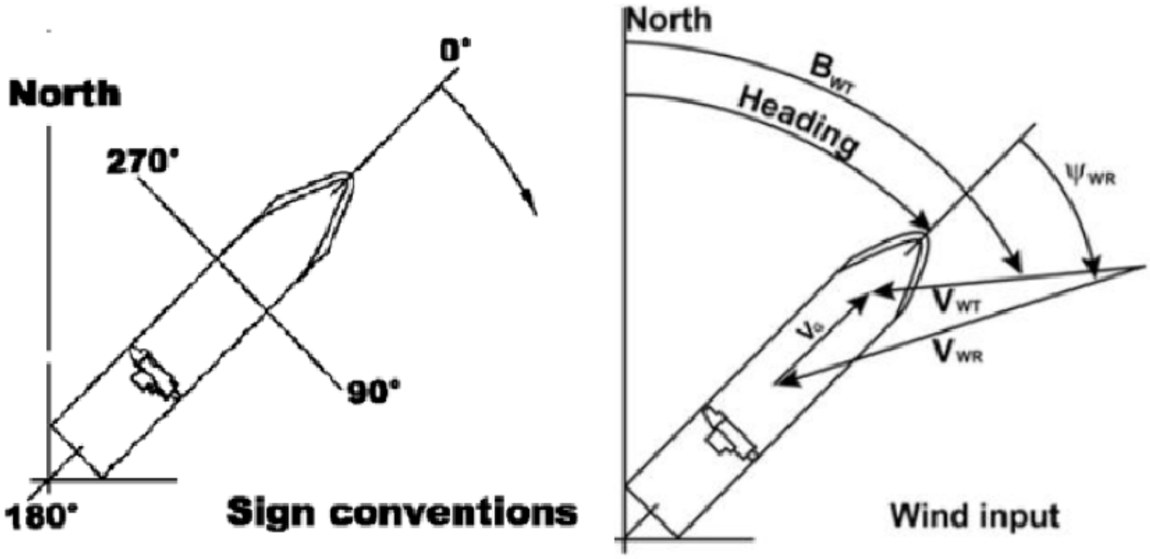


Figure D.1: To the left: Sign convention. To the right: Relevant variables shown on the vessel.

D.2 Beaufort Scale

The Beaufort wind force scale can be seen in Table D.1.

Beaufort Number	Wind speed [kts]	Wave height [m]	Description
0	< 1	0	Calm
1	1-3	0-0.2	Light Air
2	4-6	0.2-0.5	Light breeze
3	7-10	0.5-1	Gentle breeze
4	11-16	1-2	Moderate breeze
5	17-21	2-3	Fresh breeze
6	22-27	3-4	Strong breeze
7	28-33	4-5.5	High wind
8	34-40	5.5-7.5	Gale
9	41-47	7-10	Strong/severe gale
10	48-55	9-12.5	Storm
11	56-63	11.5-16	Violent storm
12	≥ 64	≥ 14	Hurricane Force

Table D.1: Beaufort Scale as described by Wikipedia (2017).

Appendix E

Regression Analysis

This appendix lists tables of coefficients, plots and regression stats that were not included in Chapter 5.

E.1 Least-Squares Regression Coefficients

Regression coefficients from the 5 most important variables for the multidimensional non-linear model.

β	**Value**
β_0	15.765
β_{t1}	-24.13
β_{t2}	133.17
β_{t3}	-160.21
β_{t4}	59.75
β_{sog4}	0.326

Table E.1: Regression coefficients from variable subset selection for multidimensional non-linear model.

Regression coefficients from the real-world data polynomial regression. A total of 25 coefficients are calculated, one for each variable, one for each variable squared and one intercept.

Variable	β	**Value**
Intercept	β_0	8.915
Atmospheric temp	β_1	0.495
Speed over ground	β_2	-52.573
Speed through water	β_3	-1.929
Sea water temp	β_4	0.341
Water depth	β_5	-0.057
Total shaft power	β_6	90.785
Total mg power	β_7	-8.795
Average draft	β_8	4.179
Days since drydock	β_9	-0.306
Days since propeller polish	β_{10}	-0.361
Total shaft torque	β_{11}	9.605
Total shaft speed	β_{12}	4.855
Atmospheric temp squared	β_{13}	-0.593
Speed over ground squared	β_{14}	26.548
Speed through water squared	β_{15}	0.964
Sea water temp squared	β_{16}	-0.155
Water depth squared	β_{17}	-0.0158
Total shaft power squared	β_{18}	-27.817
Total mg power squared	β_{19}	8.653
Average draft squared	β_{20}	-3.262
Days since drydock squared	β_{21}	-0.076
Days since propeller polish squared	β_{22}	0.108
Total shaft torque squared	β_{23}	-29.695
Total shaft speed squared	β_{24}	-23.444

Table E.2: Regression coefficients from variable subset selection for real-world.

Regression coefficients from optimal subset selection on the real-world data of size $k = 9$. All 12 variables are used as input variables in least squares regression (Table E.3).

Variable	β	**Value**
Intercept	β_0	8.915
Atmospheric temp	β_1	0.219
Speed over ground	β_2	-25.265
Water depth	β_5	-0.495
Total shaft power	β_6	6.278
Total mg power	β_7	-13.719
Average draft	β_8	0.348
Days since drydock	β_9	-0.986
Total shaft torque	β_{11}	34.481
Total shaft speed	β_{12}	-2.515

Table E.3: Regression coefficients from variable subset selection for real-world.