Data-driven Ship Propulsion modeling with applications in the Performance Analysis and Fuel Consumption prediction

PAVLOS KARAGIANNIDIS

Diploma Thesis

National Technical University of Athens

School of Naval Architecture and Marine Engineering

Supervisor: Professor Georgios Zaraphonitis

Committee member: Professor Konstantinos Spyrou

Committee member: Assistant Professor Nikos Themelis

Athens, July 2019

[This page is intentionally left blank]

Acknowledgements

The present work is the final requirement for the fulfillment of my studies in the school of Naval Architecture and Marine Engineering of the National Technical University of Athens. Therefore, and first of all, I would like to express my sincere thanks to all the teaching staff of the school and especially to my supervisor Prof. Georgios Zaraphonitis for the time he dedicated to guide me in this work, Ass. Prof. Nikos Themelis for his pieces of advice and our excellent cooperation, Prof. Konstantinos Spyrou and Ass. Prof. George Papalambrou for their contribution to the realization of this thesis.

Afterward, I would like to acknowledge the fundamental contribution of Prisma Electronics that provided all the necessary data for this study through the LAROS platform. It would be impossible to study the emerging subject of ship data analysis and data-driven ship propulsion modeling without an industry partner.

I should also thank the amazing open source community of Python users and developers that offer many essential libraries (matplotlib, pandas, numpy, scipy, plotly etc.) and especially the machine learning libraries of Tensorflow and Keras that made this work much simpler.

Last but not least, I wish to express my gratitude to my family and friends for their support during these five years of my studies, and to my fellow students with whom we had been sitting and working, side-by-side, and they have been an inspiration and a challenge for me, to become a better engineer.

Περίληψη

Η τελευταία δεκαετία στην ναυτιλία έχει χαρακτηριστεί από τις προσπάθειες για πιο αποδοτικά και φιλικά προς το περιβάλλον πλοία. Πρόκειται για μια αναγκαία επιδίωξη από τους εφοπλιστές και τους διαχειριστές των πλοίων, προκειμένου να επιβιώσουν από τις περιόδους ύφεσης της αγοράς και να συμμορφωθούν με τις αυξανόμενες απαιτήσεις των κανονισμών. Επίσης, προσπάθειες αντιμετώπισης της κλιματικής αλλαγής, σε παγκόσμιο επίπεδο, ξεκίνησαν για πρώτη φορά. Για αυτούς τους λόγους η ναυτιλιακή βιομηχανία έχει μεγάλες προκλήσεις και ευθύνες μπροστά της. Σκοπός της παρούσας μελέτης είναι η εγκαθίδρυση μεθόδων για την αποτελεσματική προ-επεξεργασία των επιχειρησιακών δεδομένων των πλοίων και η δημιουργία μοντέλων πρόωσης πλοίων βασιζόμενα σε δεδομένα, που θα βρίσκονται στον πυρήνα εφαρμογών που αποσκοπούν στη μείωση του αποτυπώματος άνθρακα των πλοίων. Έτσι, αναπτύσσονται δύο εφαρμογές τεχνητών νευρωνικών δικτύων (ΤΝΔ) που αξιοποιούν τα επεξεργασμένα δεδομένα του πλοίου, τα οποία συλλέχθηκαν αυτόματα και με υψηλή συχνότητα δειγματοληψίας, για μια περίοδο 1,5 χρόνου. Η πρώτη εφαρμογή προβλέπει τη συνολική κατανάλωση καυσίμου του πλοίου κάτω από διάφορα σενάρια λειτουργίας ενώ η δεύτερη εφαρμογή επικεντρώνεται στην παρακολούθηση της απόδοσης του πλοίου και εκτιμά την μέση απώλεια ταχύτητας του πλοίου κατά τη διάρκεια ενός έτους. Οι απαραίτητοι στατιστικοί υπολογισμοί και αλγόριθμοι για την επεξεργασία δεδομένων εφαρμόστηκαν στη γλώσσα προγραμματισμού Python και επίσης εφαρμόστηκαν οι πιο σύγχρονες τεχνικές βαθιάς εκμάθησης (deep learning) για την εκπαίδευση και τη βελτιστοποίηση των ΤΝΔ. Τα αποτελέσματα δείχνουν ότι με ένα σωστό στάδιο φιλτραρίσματος και προετοιμασίας των δεδομένων (προ-επεξεργασία), όπως αυτό που εφαρμόστηκε στην παρούσα μελέτη, είναι δυνατό να επιτευχθεί βελτίωση της απόδοσης των μοντέλων πρόωσης πλοίων και κατά συνέπεια να αυξηθεί η επίγνωση της κατάστασης του πλοίου όσον αφορά την ενεργειακή του απόδοση. Έπειτα θα μπορούν να ληφθούν πιο αποτελεσματικές αποφάσεις σχετικά με τις στρατηγικές και τα επιχειρησιακά μέτρα για τη μείωση της κατανάλωσης πετρελαίου και των αέριων εκπομπών από το πλοίο. Η κατανάλωση καυσίμου στην κύρια μηχανή εκτιμάται αρχικά με ακρίβεια 95,9% και μια βελτίωση περίπου 3% επιτεύχθηκε, μετά την κατάλληλη προ-επεξεργασία των δεδομένων, οδηγώντας τελικά σε μια μέση ακρίβεια μοντέλου 98,7%. Παρόμοια αποτελέσματα ελήφθησαν όταν άλλες παράμετροι πρόωσης (ισχύς στον άξονα, ταχύτητα πλοίου) εκτιμήθηκαν από τα μοντέλα μας. Εκ τούτου, με τόσο ακριβή μοντέλα για την εκτίμηση των παραμέτρων πρόωσης, μπορούμε να έχουμε βελτιωμένες ποσοτικές πληροφορίες σχετικά με την επιτευχθείσα μείωση εκπομπών από λειτουργικά μέτρα όπως βελτιστοποίηση διαδρομής ή ταχύτητας, και δρομολόγηση με βάση τις μετεωρολογικές συνθήκες.

Abstract

The last decade in shipping has been characterized by efforts for more efficient and environmental-friendly ships. in order to survive the downturns of the market and to comply with the increasing regulatory requirements. This is a necessary endeavor by shipowners and ship managers, in order to survive the downturns of the market and to comply with the increasing regulatory requirements. Also, efforts on the global scale against climate change initiated for the first time. That is why the shipping industry has great challenges and responsibilities lying ahead. The purpose of this study is to establish methods for effective pre-processing of ship operational data and to create data-driven ship propulsion models that will be in the core of applications that aim to reduce the carbon footprint of the ships. Thereby, two applications of Artificial Neural Networks (ANN) are developed that utilizes the processed ship data, which were automatically collected with high-frequency sampling rate, over a period of 1.5 years. The first application predicts the ship's total fuel oil consumption under various scenarios of operation while the second application focuses on the monitoring of the ship's performance and estimates the average speed loss of the ship over the period of one year. The necessary statistical calculations and algorithms for data processing were implemented in Python programming language and state-of-the-art deep learning techniques for training and optimizing Feed-Forward Neural Networks (FNNs) were applied. The results show that with a proper data filtering and preparation stage (pre-processing), like the one implemented in this study , it is possible to achieve an increased performance of the ship propulsion models and consequently increase our awareness of the ship's performance condition and take more effective decisions regarding strategies and operational measures for reducing fuel oil consumption and emissions. The main engine's fuel oil consumption was initially predicted with 95.9% accuracy and a 3% improvement was achieved after the proper pre-processing of the data, leading to a final 98.7% average model accuracy. Similar results were obtained when other propulsion parameters (shaft power, speed) were estimated by our models. With such accurate models for propulsion parameters estimation, we can have improved quantitative information on the achievable emissions abatement from operational measures like route or speed optimization and weather routing.

Table of contents

1 Introduction 9
1.1 Essence of Ship Performance 10
1.2 International Regulations & Standards regarding Ship Emissions and Efficiency 14
1.3 Literature review 17
1.4 Purpose and structure of the study 21
2 Data Acquisition 23
3 Data pre-processing 27
3.1 Data Visualization 27
3.2 Data Correction 38
3.3 Outliers Detection 43
3.4 Data Smoothing 49
3.5 Data Quality Control 52
4 Feature Engineering 56
4.1 Feature Generation 57
4.2 Feature Selection 60
4.3 Principal Component Analysis 64
5 Artificial Neural Networks: Implementation and Optimization 73
5.1 Parameters' Selection, Training Results and Optimization 73
6 Applications of ANN models on ship propulsion problems 79
6.1 Application I: Prediction of the main engine's total fuel oil consumption 79
6.2 Application II: Average Speed Loss estimation 83
7. Conclusions and Recommendations 85
References and Bibliography 86
Appendix I- Neural Networks 89

List of Figures

FIGURE 1.1 THE SHIP'S ENERGY FLOW DIAGRAM. 12
FIGURE 1.2 MAIN COMPONENTS OF THE SHIP'S TOTAL RESISTANCE. 13
FIGURE 1.3 SCHEMATIC REPRESENTATION OF THE FOLLOWED PROCEDURE FOR THE DATA-DRIVEN SHIP PROPULSION MODELING. 22
FIGURE 3.1 PLOT OVER TIME FOR THREE MEASURED PARAMETERS, ON A 2-MONTHS PERIOD OF SHIP'S OPERATION. 28
FIGURE 3.2 PLOT OVER TIME FOR THREE MEASURED PARAMETERS, ON A 1-WEEK PERIOD OF SHIP'S OPERATION. 29
FIGURE 3.3 PLOT OVER TIME FOR 3 MEASURED PARAMETERS, ON A 1-DAY PERIOD OF SHIP'S OPERATION. 30
FIGURE 3.4 SHIP'S PATH ON A WORLD MAP PROJECTION WITH (RELATIVE) TIMESTAMPS ON. EACH NUMBER ON THE MAP REFERS TO THOUSAND MINUTES. 31
FIGURE 3.5 PARAMETRIC PLOT OF STW VS PROPELLER SHAFT RPM. 35
FIGURE 3.6 PARAMETRIC PLOT OF PROPELLER SHAFT RPM VS M/E FOC 35
FIGURE 3.7 PARAMETRIC PLOT OF DRAFT VS M/E FOC 35
FIGURE 3.8 PARAMETRIC PLOT OF DRAFT VS TRIM 35
FIGURE 3.9 PARAMETRIC PLOT OF STW VS WIND SPEED 35
FIGURE 3.10 PARAMETRIC PLOT OF PROPELLER SHAFT POWER VS M/E START AIR PRES. 35
FIGURE 3.11 THREE-DIMENSIONAL SCATTER PLOT OF SHIP'S SPEED VS DRAFT VS TRIM. THE COLOR SCALE IS SET PROPORTIONAL TO THE SHIP'S SPEED FOR CLARITY. 37
FIGURE 3.12 THREE-DIMENSIONAL SCATTER PLOT OF POWER VS SPEED VS DRAFT. THE COLOR SCALE ESCRIBES THE WIND SPEED, WITH RANGE FROM 0 TO 25M/S (BLUE TO RED). 37
FIGURE 3.13 THREE-DIMENSIONAL SCATTER PLOT OF POWER VS DRAFT VS WIND SPEED WITH PARAMETER THE SHIP'S SPEED (STW). EACH COLOR CORRESPONDS TO A SPECIFIC SPEED, ACCORDING TO THE LEGEND (TOP-RIGHT). 37
FIGURE 3.14 THE LATITUDE SIGNAL AS ACQUIRED FROM THE GPS ON-BOARD THE SHIP. RAW-DATA TIME PLOT. 38
FIGURE 3.15 THE LATITUDE AND LONGITUDE COORDINATES AS ACQUIRED FROM THE GPS ON-BOARD THE SHIP. RAW-DATA MAP PROJECTION. 39
FIGURE 3.16 THE LATITUDE SIGNAL PLOTTED OVER TIME, AFTER THE CORRECTION OF OUTLYING VALUES. 40
FIGURE 3.17 THE GPS SIGNAL PROJECTED ON THE MAP, AFTER THE CORRECTION OF THE PROBLEMATIC VALUES. 40
FIGURE 3.18 SHIP'S PROPELLER SHAFT POWER MEASUREMENTS PLOTTED FOR A PERIOD OF 50 MINUTES. THE BLUE DOTS REPRESENT THE ORIGINAL DATA POINTS AND THE RED CURVE IS THE PARAMETER'S SIGNAL AFTER THE CORRECTION OF THE DATA. 42
FIGURE 3.19 SHIP'S STW MEASUREMENTS PLOTTED FOR A PERIOD OF 50 MINUTES. THE BLUE DOTS REPRESENT THE ORIGINAL DATA POINTS AND THE RED CURVE IS THE PARAMETERS SIGNAL AFTER THE CORRECTION OF THE DATA. 42
FIGURE 3.20 MAIN ENGINE'S FOC MEASUREMENTS PLOTTED FOR A PERIOD OF 1 DAY. THE BLUE DOTS REPRESENT THE ORIGINAL DATA POINTS AND THE RED CURVE IS THE PARAMETERS SIGNAL AFTER THE CORRECTION OF THE DATA. 42
FIGURE 3.21 SHIP'S LONGITUDE GPS MEASUREMENTS PLOTTED FOR A PERIOD OF 1 DAY. THE BLUE DOTS REPRESENT THE ORIGINAL DATA POINTS AND THE RED CURVE IS THE PARAMETERS SIGNAL AFTER THE CORRECTION OF THE DATA. 42
FIGURE 3.22 AN EXAMPLE OF A FALSE NEGATIVE IN THE OUTLIER DETECTION PROCEDURE. 44
FIGURE 3.23 HISTOGRAMS OF PROPELLER'S SHAFT TORQUE VALUES FOR CERTAIN RANGE OF RPM VALUES. THE "OUTLIER THRESHOLD" IS PLOTTED FOR DIFFERENT VALUES OF THE FACTOR K. 45
FIGURE 3.24 HISTOGRAMS OF M/E FOC VALUES FOR CERTAIN RANGE OF RPM VALUES. THE "OUTLIER THRESHOLD" IS PLOTTED FOR DIFFERENT VALUES OF THE FACTOR K. 46
FIGURE 3.25 THE PROPELLER'S SHAFT TORQUE, ME FOC AND STW VALUES PLOTTED AGAINST THE SHAFT'S RPM, RESPECTIVELY. IN RED COLOR ARE THE DATA POINTS THAT WERE IDENTIFIED AS OUTLIERS. IN THE ME FOC AND STW PLOTS ARE ALSO WITH RED COLOR (REJECTED) THE DATA POINT. 47

FIGURE 3.26 THE DIFFERENCE BETWEEN THE MEAN VALUE OF THE ORIGINAL DATA AND THE AVERAGED OVER DIFFERENT TIME-WINDOWS DATA. ONLY THREE BASIC PARAMETERS OF SHIP'S PROPULSION ARE PLOTTED, FOR SPACE ECONOMY. 50
FIGURE 3.27 THE MEAN STANDARD DEVIATION ON EACH DIFFERENT AVERAGING TIME WINDOW FOR THREE BASIC PARAMETERS OF SHIP'S PROPULSION. 51
FIGURE 3.28 A 5-HOUR SAMPLE OF THE SHIP'S STW PLOTTED ON ITS ORIGINAL FORM (BLUE DASHED-LINE) AND AFTER BEING SMOOTHED (RED CONTINUOUS LINE) WITH THE SMA ALGORITHM ON A 5-MINUTE AVERAGING TIME WINDOW... 51
FIGURE 3.29 THE PERCENTAGES OF THE AFFECTED BY THE CORRECTING ALGORITHM DATA POINTS FOR TWO SUBSETS OF DIFFERENT SIZE. 53
FIGURE 3.30 THE DIFFERENCE IN THE MEAN VALUE BETWEEN THE ORIGINAL AND THE CORRECTED DATASET. THE DIFFERENCES ARE ESTIMATED AS PERCENTAGE OF THE ORIGINAL DATASET'S MEAN VALUE. 54
FIGURE 4.1 FEATURE GENERATION DIALOGUE FROM THE INTERACTIVE FEATURE GENERATION PROGRAM THAT GENERATES AND STORES THE DESIRED NEW FEATURE. THE EXAMPLE OF GENERATING THE "WIND EFFECT" PARAMETER. WITH BLUE COLOR ARE THE USER'S INPUTS. 58
FIGURE 4.2 PRINCIPAL COMPONENTS ANALYSIS SEEKS A SPACE OF LOWER DIMENSIONALITY, KNOWN AS THE PRINCIPAL SUBSPACE AND DENOTED BY THE MAGENTA LINE, SUCH THAT THE ORTHOGONAL PROJECTION OF THE DATA POINTS (RED DOTS) ONTO THIS SUBSPACE MAXIMIZES THE VARIANCE OF THE PROJECTED POINTS (GREEN DOTS). AN ALTERNATIVE DEFINITION OF PCA IS BASED ON MINIMIZING THE SUM-OF-SQUARES OF THE PROJECTION ERRORS, INDICATED BY THE BLUE LINES. (SOURCE: BISHOP,2008).. 64
FIGURE 4.3 THE AXES X_1 AND X_2 ARE THE ORIGINAL FEATURE AXES, AND PC1 AND PC2 ARE THE MAXIMUM VARIANCE AXES OR THE PRINCIPAL COMPONENTS. (SOURCE: RASCHKA,2015) 65
FIGURE 4.4 THE ORIGINAL DATA POINTS OF THE PROPELLER SHAFT POWER AND PROPELLER SHAFT RPM ARE PLOTTED AGAINST THE PROJECTED ON A LOWER DIMENSION SPACE, DATA POINTS. THE NUMBER OF PRINCIPAL COMPONENTS USED FOR THE PROJECTION ARE 5 ON THE TOP LEFT CORNER AND INCREASE BY ONE AS WE MOVE LEFT AND DOWN... 70
FIGURE 4.5 THE CUMULATIVE EXPLAINED VARIANCE BY THE PRINCIPAL COMPONENTS IS A CURVE THAT PROVIDES USEFUL INFORMATION REGARDING THE POSSIBILITY OF REDUCING THE DIMENSION OF THE DATASET WITHOUT LOSS OF VALUABLE INFORMATION. 71
FIGURE 4.6 THE PRINCIPAL COMPONENTS OF TWO PAIRS OF PARAMETERS. THE ARROWS DIRECTIONS ARE THE EIGENVECTORS AND THEIR MAGNITUDES ARE THE EIGENVALUES OF THE CORRELATION MATRIX OF EACH PAIR OF PARAMETERS......... 72
FIGURE 5.1 THE RELATIVE ERROR ON THE VALIDATION SET. EACH DATASET ON THE X-AXIS TRAINS A NEW MODEL AND RELATIVE ERROR AMONG THE ESTIMATED AND THE TARGET VALUES IS THE Y-AXIS VALUE. 78
FIGURE 5.2 THE ACCURACY OF THE MODELS' ESTIMATIONS ON THE VALIDATION SET. EACH DATASET ON THE X-AXIS TRAINS A NEW MODEL AND THE RESULTING ACCURACY AMONG THE ESTIMATED AND THE TARGET VALUES IS THE Y-AXIS VALUE. 78
FIGURE 6.1 THE SHIP'S PATH IN THE TEST SET'S DATA POINTS. THE TRANSATLANTIC TRIP BEGINS FROM THE EUROPEAN SIDE (YELLOW PART OF THE CURVE) AND ENDS BACK THERE (GREEN PART OF THE CURVE). ITS DURATION WAS ABOUT 23 DAYS.................... 80
FIGURE 6.2 THE MODEL'S ESTIMATION AND THE ACTUAL ME FOC ON EACH TIME STEP (5-MINUTE) OF A TRANSATLANTIC ROUND-TRIP (TEST SET)...................................... 81
FIGURE 6.3 THE ESTIMATED ME FOC VALUE AND THE 99% CONFIDENCE INTERVAL SURROUNDING IT. 82
FIGURE 6.4 ESTIMATED FUEL COSTS AND TRIP DURATION WITH RESPECT TO THE REFERENCE SCENARIO OF AVERAGE SPEED 17.75 KNOTS... 82
FIGURE 6.5 ESTIMATION OF THE SHIP'S AVERAGE SPEED LOSS VIA TWO DIFFERENT NORMALIZATION METHODS. 84
FIGURE 0.1 AN ILLUSTRATION OF A BIOLOGICAL NEURON WITH SOME OF ITS BIOLOGICAL FEATURES THAT RELATE TO THE MODEL OF PERCEPTRON, INTRODUCED BY MCCULLOCH AND PITTS AND LATER FORMULATED BY ROSENBLATT. (SOURCE: RASCHKA,2015) 89

FIGURE 0.2 SCHEMATIC OF ROSENBLATT'S PERCEPTRON. THE BASIC ELEMENTS CONSISTING THE PERCEPTRON'S MODEL ARE THE INPUTS, THE WEIGHTS, THE NET INPUT AND THE ACTIVATION FUNCTION AND THE OUTPUT. UNTIL TODAY THEY HAVE NOT CHANGE. SOURCE: (RASCHKA, 2015)........ 90
FIGURE 0.3 NETWORK DIAGRAM FOR THE TWO-LAYER NEURAL NETWORK CORRESPONDING TO (5.5). THE INPUT, HIDDEN, AND OUTPUT VARIABLES ARE REPRESENTED BY NODES, AND THE WEIGHT PARAMETERS ARE REPRESENTED BY LINKS BETWEEN THE NODES, IN WHICH THE BIAS PARAMETERS ARE DENOTED BY LINKS COMING FROM ADDITIONAL INPUT AND HIDDEN VARIABLES X_0 AND Z_0. ARROWS DENOTE THE DIRECTION OF INFORMATION FLOW THROUGH THE NETWORK DURING FORWARD PROPAGATION. SOURCE: (BISHOP, 2006)........ 92
FIGURE 0.4 GEOMETRICAL VIEW OF THE ERROR FUNCTION E(W) AS A SURFACE SITTING OVER THE WEIGHT SPACE. POINT W_A IS A LOCAL MINIMUM AND W_B IS THE GLOBAL MINIMUM. AT ANY POINT W_C, THE LOCAL GRADIENT OF THE ERROR SURFACE IS GIVEN BY THE VECTOR ∇E. (SOURCE: BISHOP,2008)........ 96

List of Tables

TABLE 1.1 SUMMARY OF HYBRID MODELS' PERFORMANCE METRICS FROM (ALDOUS,2015). 20
TABLE 2.1 THE MAIN PARTICULARS OF THE SHIP THAT PROVIDED THE OPERATIONAL DATA FOR THIS STUDY. 23
TABLE 2.2 THE RECORDED PARAMETERS THAT ARE INCLUDED IN THE AVAILABLE DATASET. 24
TABLE 2.3 THE MEASURING DEVICES THAT ARE REQUIRED ONBOARD THE SHIP TO MONITOR THE PARAMETERS OF INTEREST. 24
TABLE 3.1 THE EXPECTED CORRELATION (BASED ON THE THEORY) BETWEEN THE AVAILABLE PARAMETERS. 33
TABLE 3.2 THE CALCULATED VALUE OF THE LINEAR CORRELATION COEFFICIENT FOR SEVEN CASES FROM THE DISCUSSED PARAMETERS 34
TABLE 3.3 TWENTY CONSECUTIVELY RECORDED VALUES OF THE SHIP'S STW (KNOTS)........ 50
TABLE 3.4 THE PERCENTAGE OF DATA POINTS THAT WERE DROPPED FROM THE DATASET AS OUTLIERS. 54
TABLE 4.1 THE LABELS OF THE ORIGINAL FEATURES OF THE DATASET. 57
TABLE 4.2 THE LABELS OF THE NEWLY GENERATED FEATURES. THE "STD" PRIOR TO THE NAMES OF THE FEATURES STANDS FOR THE TERM "STANDARD DEVIATION"........ 57
TABLE 4.3 THE LABELS OF THE INITIAL AND THE GENERATED FEATURES........ 59
TABLE 4.4 TABLE OF LINEAR CORRELATION COEFFICIENT VALUES BETWEEN INPUT FEATURES AND TARGET VARIABLES. 61
TABLE 4.5 DESCRIPTION OF THE DATA IN EACH FEATURE VIA THE CALCULATION OF A SET OF STATISTICS........ 62
TABLE 4.6 THE COMPUTATIONAL TIME REQUIRED FOR APPLYING PCA ON DATASETS OF DIFFERENT SIZE........ 71
TABLE 5.1 SUMMARY TABLE OF THE TESTING RESULTS. 77
TABLE 6.1 INPUT FEATURES FOR THE ME FOC MODEL. 79
TABLE 6.2 SUMMARY OF RESULTS OF THE ME FOC ESTIMATION APPLICATION........ 80
TABLE 6.3 THE TOTAL ME FOC FOR THE VARIOUS SCENARIOS OF REDUCED OR INCREASED AVERAGE SPEED DURING VOYAGE........ 82
TABLE 6.4 COMPARATIVE TABLE OF ISO 19030 FILTERING REQUIREMENTS VERSUS THE PROPOSED METHOD........ 83

1 Introduction

For more than 4,000 years, ships are the largest transportation means that humanity possess. It all began with flat pieces of wood, tighten together and floating on the surface of the sea or a lake. It was a tool for satisfying the curiosity or the survival needs of some group of people living by the shore. Primitive boats had a catalytic role in the spread of our species around the globe and to our survival and dominance over every possible obstacle on earth. With the help of a boat, you could flee from a hostile environment, avoid famine and find a proper place for the continuation of human life.

In the modern world ships still have the same significance but from a new perspective. They serve, less primitive but equally fundamental concepts of the human society. World trade and Globalization are mainly supported by the shipping industry. People were carrying different forms of capital with them when sailing for commerce or migration, not long ago, since ships were the only mean of intercontinental transportation. Human capital, not only in the form of labor but also as knowledge and technical know-how, and Social capital are some examples of what was transferred by the ships. Capitalism would not have been able to flourish in the last centuries if it was not for the ships to implement the economic theories in practice. Free trade and its benefits, like "Comparative Advantage" have been pushing economic growth for decades because commercial ships are able to transfer large quantities of goods with very low cost per unit transferred. This last characteristic of commercial ships is the key to the shipping industry's future.

However, in order to achieve this economy of scale, ships have nowadays become huge floating factories. They consist of numerous machinery equipment, engines and networks of pipes or cables. Dozens of people live and work onboard the ships and very delicate cargos, worth of millions of dollars, are being transferred by them. In addition, numerous experienced engineers, operators and managers are employed on-shore in order to ensure the successful and efficient transfer of these cargoes by the modern ships. As a result, the performance of such a complex and significant engineering construction, as the ship, has come under the microscope of the scientist and engineers.

The present study is an attempt to investigate the field of propulsion modelling for the scope of performance analysis and monitoring. Focus is given in the recently introduced and promising data-driven technics for propulsion modelling and predictive analytics. New capabilities are offered to the researchers in this field, due to the installation of automatic data acquisition systems onboard the ships, in the last decade. However, before addressing the issue of propulsion modelling, we should briefly discuss certain important concepts that are related to the scopes of the study.

In this introductory section we shall describe the concepts of ship's efficiency and performance, and the reasons why it is so essential nowadays to pursue high efficiency and optimum performance. Also, we shall review the related literature on the topics of data-driven ship propulsion modelling and ship performance and efficiency.

1.1 Essence of Ship Performance

Ship Performance is a quite abstract term. It would be difficult to give a clear definition with just a few words. That is why we should define the framework over which we wish to use this term. So, before becoming any more specific, let us have a look on the greater picture of the ship's operation and on how efficiency relates with performance. Also, note that our analysis is valid mainly for large merchant ships.

Usually in the fields of science, engineering or finance any term involving the words performance or efficiency is a fraction where the denominator is the "what we give" and the nominator the "what we receive". In this sense, the efficiency of a heat engine is the produced work over the total energy consumption or the performance of an investment is the monetary value of our absolute gain (or loss) over the monetary value of the invested capital. Consequently, the propulsive efficiency of a particular ship on operation could be defined as the total distance travelled over the total consumed energy, in a specific time period. Alternatively, in order to avoid testing specific time periods, it could be the ship's propulsive power over the total power consumption at the instant. Finally, the term "ship's performance" could be regarded as the ship's propulsive efficiency over time, because the ship's efficiency is a design parameter but the ship's performance is an operational parameter.

i. The flow of the energy in ships

Assuming the above definitions for efficiency and performance, the terms "propulsive power" and "total consumed energy" should be analyzed. Firstly, the total consumed energy is the energy that we offered to the ship, as a physical system, and the propulsive power is the actual work that we are able to retrieve from the system. However, by estimating the ships efficiency, we see only the two ends of an interesting and complicated process. It is the flow and transformation of energy through the various mechanical systems of the ship that ultimately define the ship's efficiency. In order to comprehend and improve efficiency and performance, the energy flow through these systems should be investigated.

In fact, there is a single energy source (input), for the majority of ship types and this fact simplifies considerably our analysis. The journey of the energy initiates with the bunkering procedure, where the ship receives chemical energy in the form of fuel oil (HFO, MDO, LNG etc.). This fuel oil is the sole energy source of the ship and therefore it is equal to the total consumed energy.

The reason for this, is that the heart of the ships is the diesel engine, since it has dominated over all alternative options from the beginning of the 20th century and until today. The diesel engine is responsible for the transformation of the chemical energy to heat and then to mechanical work. Obviously, during these transformations, there are remarkable energy losses, which are the amount of energy that escaped from the physical system without offering any valuable work. In a large and modern diesel engine, the total energy losses are about 46% and it is mainly energy in the form of heat that is wasted due to friction, radiation or for cooling needs. However, this amount of wasted energy is inevitable and there can only be marginal improvements in the efficiency of the diesel engines in the future.

In regards to the remaining mechanical power, that is devoted to the rotation of the ship's shafting system, additional heat losses appear due to friction between the shafts, the lubricant and the bearings' surfaces. The percentage of energy losses here depends on the size and type of the ship. Fast ships usually have thinner hull shape and consequently longer shafting system with more bearings that result in higher frictional losses. Advances in the field of tribology are improving the total efficiency of the shafting system but again the margins for improvement are tight. Also, in the case of RoPax ferries, 4x stroke Otto engines are commonly used and besides the reduced thermodynamic efficiency of these engines, they include a reducer in the shafting system that exhibits additional heat losses due to friction. In summary, the energy losses from the crankshaft till the stern tube of the ship are about ~1% to 3% of the engine's power output.

The final step of the energy journey is the propeller. There, the mechanical energy from the shaft's rotation is absorbed by the propeller in order to move its blades through the water and produce the force required to accelerate and maintain the ship on a service speed. From this point and after, the phenomena that have to be studied are mostly hydrodynamic. Once again, energy losses appear, during the transformation of shaft's torque (kinetic energy) to thrust power. The propeller's blades function as hydrofoils that generate lift, not in the vertical direction as usual, but horizontally, in the direction of the ship's speed. It is during this procedure of generating thrust due to pressure difference in the two sides of the propeller disc, that energy losses occur.

Ideally, the produced thrust multiplied by the ship's speed, $V \cdot T$ would be equal to the torque of the shaft multiplied by its angular velocity, $Q \cdot \omega$. In such case, all the power transmitted by the shafting system would be turned to thrust power accelerating the ship through water. However, this cannot happen for two specific reasons. On the one hand we have the propeller efficiency on free flow and on the other hand the relative efficiency from propeller and hull interaction. The propeller's efficiency on free flow depends solely on its design, hence it shall be considered predetermined and its efficiency is could be around 75%. Additionally, the existence of the hull in front of the propeller, distorts the flow of the water towards the propeller by reducing the axial velocity of the flow. Also, the rotation of the propeller in the stern of the ship increases the drag of the hull by furtherly reducing the pressure in the area. The influence of these two effects on the total efficiency of the ship's propulsion is quantified by the wake fraction coefficient, *w*, for the former and the thrust deduction factor, *t*, for the latter phenomenon. Overall, it is common to have propellers with 60% efficiency which means that another 40% of the energy at this stage is lost.

One last factor that affects the ship's efficiency by increasing the hull's resistance, is the added resistance from the appendages. Appendages may be added to the bare hull for maneuvering, structural or stability reasons. The most commonly added piece of equipment that could be modified to increase efficiency is the rudder that is placed close to the propeller and interacts constantly with her. Other features of interest, may be ducts or fins that improve the hydrodynamic efficiency of the hull as a total. The w and t values are affected by this energy saving equipment, but they are also affected from the hull and/or propeller fouling that occurs over time. Cavitation phenomena also relate with energy loss and reduced propulsive efficiency but their analysis is out of the scopes of the present study.

The above analysis of the energy flow through the ship's systems is summed up in Figure 1.1. Every box in the figure describes an activity or stage in which energy is consumed without producing thrust.

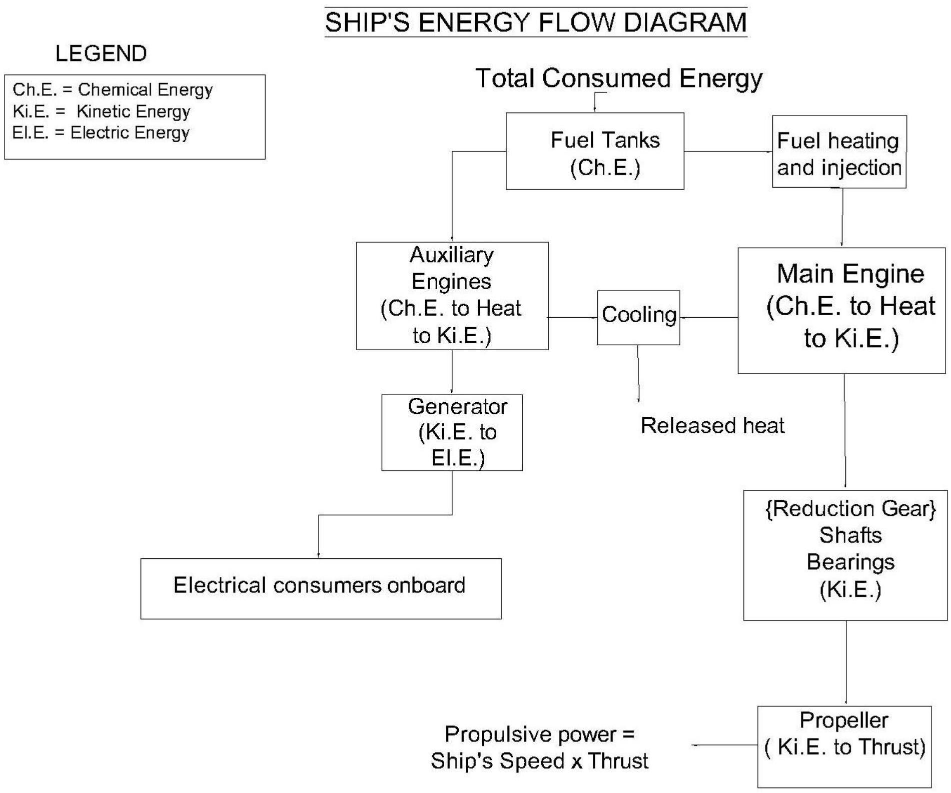

SHIP'S ENERGY FLOW DIAGRAM
LEGEND
Ch.E. = Chemical Energy
Ki.E. = Kinetic Energy
El.E. = Electric Energy
Total Consumed Energy
Fuel Tanks (Ch.E.)
Fuel heating and injection
Auxiliary Engines (Ch.E. to Heat to Ki.E.)
Cooling
Main Engine (Ch.E. to Heat to Ki.E.)
Generator (Ki.E. to El.E.)
Released heat
Electrical consumers onboard
{Reduction Gear} Shafts Bearings (Ki.E.)
Propulsive power = Ship's Speed x Thrust
Propeller (Ki.E. to Thrust)

Figure 1.1 The ship's energy flow diagram.

ii. Factors that influence the ship's performance

Comprehending the greater picture of the energy's flow through the ship's systems can lead to more systematic performance analysis and monitoring. As mentioned, ship performance is the relative ship efficiency over time and hence, optimal performance is equal to minimum reduction of the ship's efficiency. On the one hand, the ship's efficiency depends on many different subsystems, each one working on its own terms but also interacting with other subsystems of the ship. Therefore, a ship will operate at its maximum overall efficiency when it's newly delivered or properly maintained, and all of its subsystems are fine-tuned to operate on the region of their maximum efficiency. However, optimizing the ship's efficiency is not the same task as optimizing the ship's performance. That is because, on the other hand, the ship's performance depends solely on maintaining the ship's efficiency on its initial level, no matter if it operates in conditions of high or low efficiency. For this reason, on a performance monitoring task, there is no reason to involve the optimization of the machinery and equipment. Only the time-dependent parameters that affect the ship's efficiency should be studied, in order to know when and how to act to improve the performance.

As found in many studies that focus on ship performance [i.e. (Karaminas & Shen, 2016), (ISO 19030, 2016)], the main cause of poor performance is the hull and propeller fouling. With the term fouling we mean the appearance of aquatic life on the ship's surfaces. More specific, it is the accumulation of microorganisms, plants, algae, or animals on the wetted surfaces of the ship. They have a direct impact on the ship's performance because they increase the frictional resistance of the hull and result to higher power demand for the same service speed.

Additionally, the ship's efficiency is optimized for a few service conditions, the design conditions. These conditions can be described by the speed, the draft and the trim of the ship, and usually refer to calm sea. Consequently, when the ship sails in different conditions its efficiency and maybe its performance varies because its resistance varies as well. It is the objective of the performance analysis to detect, explore and quantify those variations and understand their root cause.

In Figure 1.2, the analysis of resistance in components (Taylor, 1910) contributes in the comprehension of resistance variations, due to changes in the operating and loading conditions of the ship.

The draft of the ship depends on the weight that is carried, and defines the volume of the submerged part of the hull.

The trim of the ship is defined by the distribution of the weight on the ship, and hence it can differ, for the same loaded weight.

The combination of a particular draft and trim defines the **loading condition** of the hull and therefore the shape of her submerged part and her wetted surface as well. The former affects the pressure or drag component of the total resistance, since the hydrodynamic flow around the hull is different while the latter affects the frictional component of the ship's total resistance.

The speed of the ship also influences the total resistance of the ship because the wave making component depends heavily on the ship's speed since different speeds result to different Froude number. Also, if the hull has a bulb in the bow then the bulb will operate effectively only for certain combinations of speed, trim and draft.

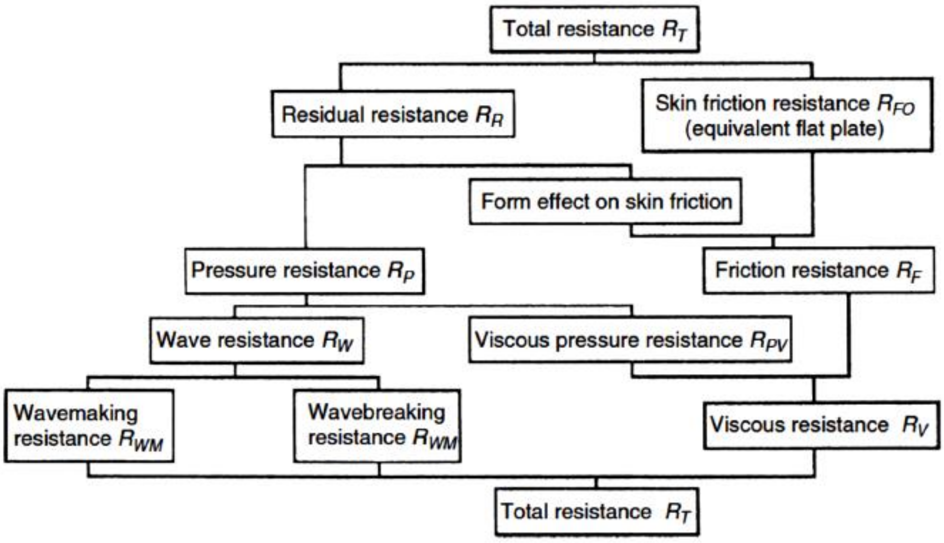

Total resistance R_T
Residual resistance R_R
Skin friction resistance R_{FO} (equivalent flat plate)
Form effect on skin friction
Pressure resistance R_P
Friction resistance R_F
Wave resistance R_W
Viscous pressure resistance R_{PV}
Wavemaking resistance R_{WM}
Wavebreaking resistance R_{WM}
Viscous resistance R_V
Total resistance R_T

Figure 1.2 Main components of the ship's total resistance.

In a performance analysis task, we are not only interested in which are the speed, draft and trim values that maximize the propulsive efficiency since these values are provided by the designer. The scope is to be able to compare, among the varying loading and environmental conditions, the efficiency of the ship's propulsion, so that any reduction, over time or for specific conditions, can be detected. This fact, constitutes the constant monitoring of the ship's propulsion parameters essential in the procedure of evaluating and improving the ship's performance.

1.2 International Regulations & Standards regarding Ship Emissions and Efficiency

The strive for higher efficiency and optimum performance in the shipping industry has been rising remarkably in the last decade. It could be considered as a strategic business decision to maximize profits by minimizing the energy consumption, and especially the energy losses in the ships. However, another crucial reason that almost all ship operators focused their efforts towards achieving a more efficient fleet, is that the legislative authorities around the shipping industry are planning for an eco-friendlier future. This ambition has brought the global fleet's gas emissions under the need for severe mitigation while the demand for transportation capacity will keep on rising (DNV-GL, 2017). This is the shipping industry's greater challenge for the near future and a motivation for the current study. Hence, we should investigate how the legislative and regulatory authorities try to impose this transition to an environmentally friendlier future.

At the highest level of international cooperation and intergovernmental agreements, there is the United Nations (UN) organization that promotes the discussion and the mutual action of all nations over global affairs. The United Nations is the organization that shows the directions and sets the goals that other international organizations or the national governments should achieve. Lately, one of the most important agreements in the UN, was the Paris Agreement that aims to slow down climate change and avoid the harsh consequences that it would bring.

The Paris Agreement is an agreement within the United Nations Framework Convention on Climate Change (UNFCCC), dealing with greenhouse-gas-emissions mitigation, adaptation, and finance, signed in 2016 by 194 states. It identifies a clear goal of "holding the increase in the global average temperature to well below 2°C above pre-industrial levels and to pursue efforts to limit the temperature increase to 1.5 °C above pre-industrial levels." As a result, all the involved parties are obligated to legislate in the direction of greenhouse gases (GHG) abatement or indicate other measures that will aim to constrain the increase in the global average temperature.

The International Maritime Organization, is an agency founded by the United Nations in 1948, and is responsible for providing the regulatory framework for the international shipping. The IMO is also responsible for the adoption and implementation of the proposed international regulations, on national level. Therefore, in consistence with the Paris Agreement, the IMO has adopted regulations that aim to control the air pollution from ships' emissions or measure the energy-efficiency of ships in order to set limits and constrain the GHG emissions.

In the first category of measures belong the prior to Paris Agreement, **Emission Control Areas** (ECAs) and the **Global Sulphur Cap** (GSC), that prevent the emission of SOx, NOx and PMs in order to improve the quality of air and protect the environment. These emissions can be constraint either by technologies that capture or prevent the formulation, of the above-mentioned oxides or by the use of alternative fuels. Hence compliance with these regulations is not achieved through operational measures, like improved performance and so we will not discuss them any further.

However, for the GHG emission (CO_2, CH_4) reduction, there can only be measures that limit the emissions in total, by improving the energy-efficiency or reducing the energy consumption of the ships. So far, there have been several regulations that aim to achieve this goal.

The MRV (Monitoring, Reporting and Verification) **regulation** was introduced in 2015, as a first, preparatory, step in the process of limiting the GHG emissions from the ships. It aims to quantify the amount of emitted CO_2 so that specific goals for reduction can be set afterwards. Also, some people in the industry believe that the MRV should be exploited to establish a carbon Emission Trading System (ETS), while other advocate for different Market Based Measures (MBMs), i.e. bunker levy (Psaraftis, 2019). Even though there is still a lot of uncertainty around these issues, the IMO has proceeded to set goals for the energy efficiency of the ships.

The major goal of the **de-carbonization** of the shipping industry is planned to be implemented in three stages. In the first stage, the requirement is to have 30% more efficient ships by 2025 and the final goals are expected to be a 40% reduction by 2030 and a 70% reduction by 2050, to the total annual GHG emissions. The baseline year in which these reductions refer to, is the 2008. These goals have been decided by IMO's Marine Environment Protection Committee (MEPC) and were based on the Second and Third IMO GHG Study (Smith, et al., 2014). Hence, IMO will have to oblige ship operators to achieve this reduction in emissions and for this purpose, additional regulations have been introduced (MEPC.62, 2011). In the Annex VI a new Chapter 4 entitled "Regulations on energy efficiency for ships", is making mandatory the Energy Efficiency Design Index (EEDI) for new ships and the Ship Energy Efficiency Plan (SEEMP) for all ships.

The **Energy Efficiency Design Index** (EEDI) is an index that estimates the emitted grams of CO_2 per transport work (tonne-mile), for each ship. This means that the smaller the value of the index, the more efficient the ship’s design. It is a function of just three elements, in order to be simple and capable of broad application. These are:

a. the installed power,
b. the speed of vessel,
c. the cargo carried.

It is applicable only to new ships (delivered after 2012) and aims to ensure that they are designed to be more energy efficient than the already existing designs. This is achieved by defining a reference line (baseline) and requiring the attained EEDI value of each ship to be below this baseline. However, the attained EEDI value will have to be reduced over time until the goal of 30% more efficient ship designs is achieved by 2025 and onwards. That demand is expressed with the following formula:

$$Attained\ EEDI \leq Required\ EEDI\left(1-\frac{X}{100}\right)\cdot Baseline$$

Where X is the reduction factor that will be increasing every five years (2015,2020,2025). Also, the baseline is ship-type specific and in general is easier for large and slow ships to comply with the EEDI requirements. Nevertheless, the fact that the EEDI is not a performance indicator of the operational energy efficiency of vessels, makes it indirectly related to the objectives of this study.

The **Energy Efficiency Operational Indicator** (EEOI) on the other hand, is a tool for monitoring ship efficiency over time. It was proposed in (MEPC.1/Circ.684, 2009), as a voluntary measure. It aims to gauge the effect of any changes in operation, e.g. improved voyage planning, and more frequent propeller cleaning, or the introduction of technical measures such as waste heat recovery systems or a new propeller. It is applicable to both, existing and new ships, and it has the following basic expression:

$$EEOI = \sum_{j} \frac{FC_j \times C_{F_j}}{m_{cargo} \times D}$$

Where j is the fuel type, FC_j is the mass of the consumed fuel, C_{F_j} is the fuel mass to CO_2 mass conversion factor for fuel j, m_{cargo} is the carried cargo (tonnes or TEU or passengers) and D is the distance in nautical miles corresponding to the cargo carried. Also, average EEOI or moving average EEOI (average over a time-window) can be exploited for the same goal.

Finally, the **Ship Energy Efficiency Management Plan** (SEEMP) is the second mandatory measure proposed in (MEPC.62, 2011)that must be adopted by ship operators in order to comply with the regulations on Energy Efficiency for Ships in MARPOL Annex VI (IMO, 2012). It is an operational measure that establish a mechanism to improve the energy efficiency of a ship in a cost-effective manner. The guidance on the development of the SEEMP for new and existing ships incorporates best practices for fuel efficient ship operation, as well as guidelines for voluntary use of the EEOI. The SEEMP urges the ship owners and operators at each stage of the plan to consider new technologies and practices when seeking to optimize the performance of a ship.

Last but not least, another reason that is not directly addressed by the regulations but constitutes a significant reason for increased GHG emissions and other types of environmental pollution by the ships, is the hull and propeller fouling. Three ways that hull fouling can impact the environment are presented in (Logan, 2011).

a. Increased the GHG emission due to the extra power and fuel consumption needed to maintain service speeds.
b. Toxic paint residuals from the periodic hull cleaning can pollute the marine environment.
c. Probable transportation of aquatic invasive species that resident on the ship’s fouled hull and may damage local ecosystems or the human health and the local economy.

Therefore, the ability to detect the hull and propeller fouling in order to proceed in timely hull and propeller cleaning, is a major issue and a motivation for this study.

For the purpose of timely detection of the hull fouling and the monitoring of the ship’s performance the **ISO 19030 “Ships and marine technology – Measurement of changes in hull and propeller performance”** has been published (ISO 19030, 2016). It defines some specific procedures for the data collection and preparation and introduces five Performance Indicators (PIs). In the present study, we are not going to follow the ISO 19030 standards and procedures but we will use it a few times as a reference or comparison point.

1.3 Literature review

The subject of modelling the ship's propulsion is being studied since the beginning of the 20th century. In 1910 the Rear Admiral of the U.S. Navy, D.W. Taylor published the book "The Speed and Power of Ships" (Taylor, 1910). This work laid the foundations for the development of the ship's resistance and propulsion theory in the past century. Another landmark publication for this field is the second volume of the famous series "Principals of Naval Architecture", with title "Resistance, Propulsion and Vibrations", published by the Society of Naval Architects and Marine Engineers (SNAME), (Lewis, 1988). These two books are addressing the issue of modelling the ship's propulsion with a combination of experimental and theoretical approach that leads to models with semi-empirical equations. Examples of these type of methods are the systematic series (Wageningen, MARAD etc.) or Holtrop- Mennen method (Holtrop & Mennen, 1982). More recent works in the field, include physical modelling of the propulsion plant and solve numerically, differential equations that attempt to describe reality in greater detail than the semi-empirical equations (Theotokatos, 2007). This approach in general, is the most systematic way to address the issue of powering requirements for the ship's propulsion and is still used in design or performance analysis tasks.

However, the accuracy of this 'classical' approach is often limited (Pedersen & Larsen, 2009). The required power to propel a ship at a certain speed is rarely estimated with error smaller than 10% and if it is a novel design then the empirical equations should not even be applied. Therefore, a good alternative to these methods is the Computational Fluid Dynamics (CFD) simulations. CFDs provide much higher accuracy in the estimation of the ship's resistance but they have really high computational cost and complexity. The reason is that the system of partial differential equations (Navier- Stokes) that needs to be solved on each and every computational volume of the defined mesh, can take hours of computation on a high-end computer and it will result to the simulation of a single loading condition for the ship. Overall, we would say that they are a very useful tool for detailed estimations and simulations of phenomena of interest but they are not a convenient tool for modelling the ship's propulsive efficiency for the numerous conditions that will be faced in reality.

The need for higher accuracy than the empirical or theoretical models and lower computational cost than the CFD simulations, lead the scientific community to the experimentation with data-driven and ship-specific methods. The basic idea behind these models is to exploit the data collected from a particular ship's operation and use them to create a statistical model that could estimate its powering needs or forecast its consumption and monitor its performance. A number of papers that are applying this type of propulsion modeling are presented below.

In the PhD thesis of (Petersen & Winther, Mining of Ship Operation Data for Energy Conservation, 2011)the propulsion of a RoPax ship with two controllable pitch propellers (CPP) is modelled as a time-series problem and automatically collected, high frequency data are used. The available (collected) propulsion parameters are divided into three groups.

- The **state vector** parameters: V_{STW} (Speed Through Water), Trim, Draft, FOC (Fuel Oil Consumption), Heading.
- The **control vector** parameters: Propellers' pitch, Rudders' Angle, Longitudinal Speed Difference (V_{STW} -V_{SOG}), Head and Cross (the parallel and the perpendicular component to ship's speed vector, respectively) Wind Speed.
- The **constant vector** parameters: Initial Port and Starboard Level

Afterwards a Time-Delay Neural Network (TDNN) and a Gaussian Mixture Model (GMM) are trained on these data in order to create a model that predicts the ship's response. The input to these networks is the current values of the state, control and constant vectors and their two previous values (a time-series of three timesteps). The trained models are tested either as regression model (estimator) or as dynamic model(predictor) of the ship's response. In the first case, the estimated target's value is compared to the measured one. In second case, the predicted target's value for the next time-step is compared to the measured one. Prior to the training, the dataset is divided trip-wise and shuffled. Then one-third of the trips are used as training set, one-third as validation set and one-third as test set. During the validation phase a noise distribution (Gaussian multivariate) is fitted to the residuals of the model's estimations. After that, the FOC and the V_{STW} are forecasted for the test set data, by both models (TDNN and GMM). The results are given in the form of plots with the measured and the forecasted time-series surrounded by the 50% and 90% percentile intervals.

Unfortunately, the results are not accompanied by some loss metrics of the estimated or forecasted values versus the measured ones. Also, the scale of the plots is quite large (in y-axis) and so details of the signals are not clear. Nevertheless, we notice that the estimations fall much closer to the measured values and the percentile intervals are much narrower, when the data refer to steady-sailing conditions. In other cases, we see the forecasted signal to miss the trend or drift from the measured values and in transient cases the percentile intervals (for the GMM only) around the FOC where inconveniently large.

Again, in (Petersen, Jakobsen, & Winther, 2011) a paper part of his PhD, a publicly available dataset of high-quality sensory data, collected from a ferry over a period of two months, is used for the training of two types of neural network models. An instantaneous and a predictive model, that estimate the ship's FOC. The instantaneous, is a typical feed-forward neural network model. The predictive is a TDNN (a type of Recurrent neural network) that takes as input X_n, all the available propulsion parameters (state, control and constant vectors, except ship's heading) and estimates just the difference in the target variable at the next time-step, X_{n+1}. Also, the predictive model's residual error in the training set is used to fit a probability distribution that will be added to the model's final predictions as noise. A simple feature extraction procedure follows, where the mean the variance and the derivative of some parameters are the generated features. In the averaging of the data, a time-window size of 3 minutes is used for the instantaneous model and of 15 seconds for the predictive. The weights of the neural networks are regularized with the addition of a penalizing term in the error function, governed by the hyper-parameter λ. The value of λ is determined based on the k-folds cross validation process. The two-thirds of the data are used as the training set, which is divided again trip-wise and shuffled. The test set as well, contains data of whole trips.

The instantaneous model's predictions align well with the test's set target signal. However, when the training data are shuffled without being grouped trip-wise, the performance of the instantaneous model is further improved. The quantitative results are presented in comparison with two other studies: They achieve a mean relative error of 1,50% on the FOC prediction while (Pedersen,2009) reports 1,65%. In absolute values, the RMS (Root-Mean-Squared) error on the speed estimation is 0.32 knots and in the FOC estimation 41.1 L/h while (Leifsson et al., 2008) report RMS errors of 0.65 knots in speed and 60 L/h in the FOC.

In regards to the dynamical model, it uses a TDNN with two groups of input parameters, the control and the dynamic ones.

Control parameters	Dynamic parameters
– Port and Starboard propeller’s pitch – Port and Starboard rudder’s angle – Initial Port and Starboard level – Difference between ground and through water speed – Headwind and Crosswind	– Speed Through Water – Port and Starboard level – Trim – Draft – Difference in heading

The control parameters are used to predict the changes in the values of the dynamic parameters on each time-step. As mentioned earlier, in the model’s output a noise term is added, which is sampled from a Student’s distribution, fitted over the residuals of the validation set. The presented results are only qualitative and demonstrate the simulation of the ship’s STW as a response to a certain pitch signal.

In this study, we expected to see an argument for the selection of time-window size when averaging the data for the instantaneous and the predictive model. Also, for the proper estimation of the derivative of a parameter, the sampling frequency should be high enough, in regards to the parameter’s dynamics. However, there is no discussion about this issue before estimating the derivatives of the parameters. Last but not least, we notice that there is no information about the engine’s operation in the dataset (i.e. engine’s torque or rpm).

Another effort of modeling the ship’s propulsion with neural networks is presented in (Pedersen & Larsen, 2009). They attempt to predict the propulsion power of a 110,000-dwt tanker. A high frequency dataset has been obtained from sensors on board the ship and the data are organized to 10-minute intervals. Furthermore, the ship’s heading is used to filter the data since it was noticed that when the heading was changing there was significant influence on the measured propulsion power. The wind speed is an indicator for the wind-driven waves but information about swell is not included. Finally, the data are split into four different sets, each one with different mean draft and/or trim value.

The input vector for model is the following:

- Speed through water
- Wind speed
- Wind direction
- Air temperature
- Water temperature

For obtaining the best performing model, each neural network is trained 10 times and the k-folds (k=5) cross validation technic is applied as well. Architectures with 5, 10, 15 and 20 hidden units, in the single hidden layer of the network, are tested. The best results are achieved with the 15 and 20 hidden units and the mean relative error ranges from 0.82% to 2.69%. Results from the prediction of the same target values (propulsion power) with empirical methods are presented and their error was found to be about ten times higher.

The efforts in the data-driven modeling of the ship’s propulsion however, should not be constrained in the field of neural networks only. In the PhD thesis of (Aldous, 2015)we get an extended treatment of the ship performance issue. It is a study with a more systematic approach on the issues of data uncertainty and pre-processing, which are crucial elements of the data-driven modelling procedure that were not given the necessary attention in the previous papers. Data from noon reports (NR) are compared to continuous monitoring (CM) data and the main disadvantages of the first are found to be: (a) the lack of standardization, (b) the missing observations and inaccuracies, and (c) the inherent characteristic of expressing the ship’s performance in terms of FOC only, and not in the more appropriate terms of shaft power. The

CM dataset provides a much larger amount of data in the same time period but it's only negative attribute is that it requires frequent maintenance. Also, the onboard obtained CM data are compared with met-ocean data and are found to only partially overlay. The concluding argument is that CM datasets are more proper for the assessment of the ship's performance. This argument is found to be supported in other works too, like (Themelis, et al., 2018b).

Afterwards, there is a broad review of the proposed performance indicators (PI) in the literature, and then the author decides to use the following PI in her study:

$$P_{s,measured} - P_{s,modelled}$$

Where P_s refers to shaft power and $P_{modelled}$ would be calculated from a model, derived either from theory or during a calibration period which would form the 'training' dataset for a statistical or hybrid model.

Before proceeding to the modelling phase, an extensive discussion about the uncertainty in the data and in the models, takes place. The scope is to provide an uncertainty framework that is divided into four categories: instrument uncertainty, sampling uncertainty, model's uncertainty and for the case of NR datasets, the human error derived uncertainty. The discussion about uncertainties leads to the introduction of filtering criteria. The dataset is filtered with respect to certain parameters like water depth, shaft power, speed and sea current. Also, a type of statistical filtering is applied with respect to the FOC parameter, by comparing the measured values to the modelled ones and rejecting those that fall 2σ away from the mean. However, we believe that filtering with respect to a simple theoretical model of FOC versus Shaft Power and not taking into consideration other parameters is not optimal. Also, the temporal resolution of the available data for this study was 15 minutes, which is a common resolution for performance analysis studies but it would be preferable to be higher, like in (Pedersen & Larsen, 2009) or (Petersen et al., 2011).

The models produced in Aldous thesis were theoretical, statistical (linear regression) and hybrid (a combination of both). They are only partially related to the present study because neither linear regression or theoretical modeling is used here. However, it is very important to report the accuracy that these types of models achieve in order to justify why we choose to work on other types of data-driven models. In the case of the statistical model, for the prediction of the FOC it is reported a Root Mean Squared Deviation (RMSD) of 8,200 tonnes per day (tpd) and the R^2 value is 0.862. Respectively, the RMSD for the prediction of the Shaft Power is 1654.022 kW and the R^2 value is 0.892. In the case of the hybrid models the results are summarized in Table 1.1.

Table 1.1 Summary of hybrid models' performance metrics from (Aldous,2015).

		Model's Performance Metric	
Model	**Parameter**	RMSD	R^2
Hybrid I	FOC	6.502 tpd	0.820
	Shaft Power	1335.422 kW	0.836
Hybrid II	FOC	5.540 tpd	0.848
	Shaft Power	1141.473 kW	0.860

So far, it should be clear that this type of models cannot be as accurate as the more complex type of models (i.e. neural networks) are in the estimation of the propulsion parameters. This statement is verified again in the work of (Coraddu, et al., 2016) where the performance of Black and Gray Box Models is measured with various model performance indicators and when compared to White Box Models (the theoretical ones) is found to be superior.

1.4 Purpose and structure of the study

Even though the GHG emissions from ships are less than 3% of the global GHG emissions (Smith, et al., 2014), while they transfer more that 90% of the goods, the IMO has set goals for reducing them. Therefore, one of the main incentives of this study is to provide some model-based tools to the shipping industry, that will assist to reduce the energy consumption and the gas emissions from the ships, in order to achieve these environmental goals.

Simultaneously, the developments in different fields of technology has made feasible the continuous monitoring of ships and the gathering of huge amount of operational data, that carry precious information, if properly exploited. More specific, it was the convergence of: (i) technologies in the field of the electronics and sensors, (ii) the global satellite internet connection and lately, (iii) the machine learning methods, that produce models with superior predictive capabilities. Thereby, another incentive of this study is to utilize these newly appeared large datasets and machine learning methods in order to build useful tools. With the term tools we mean optimized, data-driven propulsion models (black box models) that can later be used in various applications, like decision-support tools or simulators.

The combination of the two aforementioned incentives, determines the ultimate purpose of this study, which can be summarized as follows: The capability to produce highly-accurate ship propulsion models, through improved technics and practices of data manipulation, will enable the thorough and effective investigation of operational measures that could lead in the reduction of the GHG emissions from the ships. Such models can be utilized in applications of route optimization, speed optimization, weather routing, and performance monitoring.

An overview of the proposed procedure for propulsion modeling in the era of automatic and high-frequency data collection is presented in Figure 1.3, which is also an overview of the present study, as well. Firstly, a brief description and some information about the sensors and the data acquisition system are given in Chapter 2. Then, the data pre-processing and quality control stages, are covered in Chapter 3. Great emphasis is given in this stage because it hides remarkable possibilities for improving the performance of the produced model. The process of feature engineering is discussed in Chapter 4. The model selection, training, and optimization are presented in Chapter 5, along with results for the models' performance that justify and support the actions of the previous Chapters (3 and 4). Finally, some applications of the produced models are presented in Chapter 6. An introduction to specific machine learning methods (artificial neural networks) is presented in Appendix A.

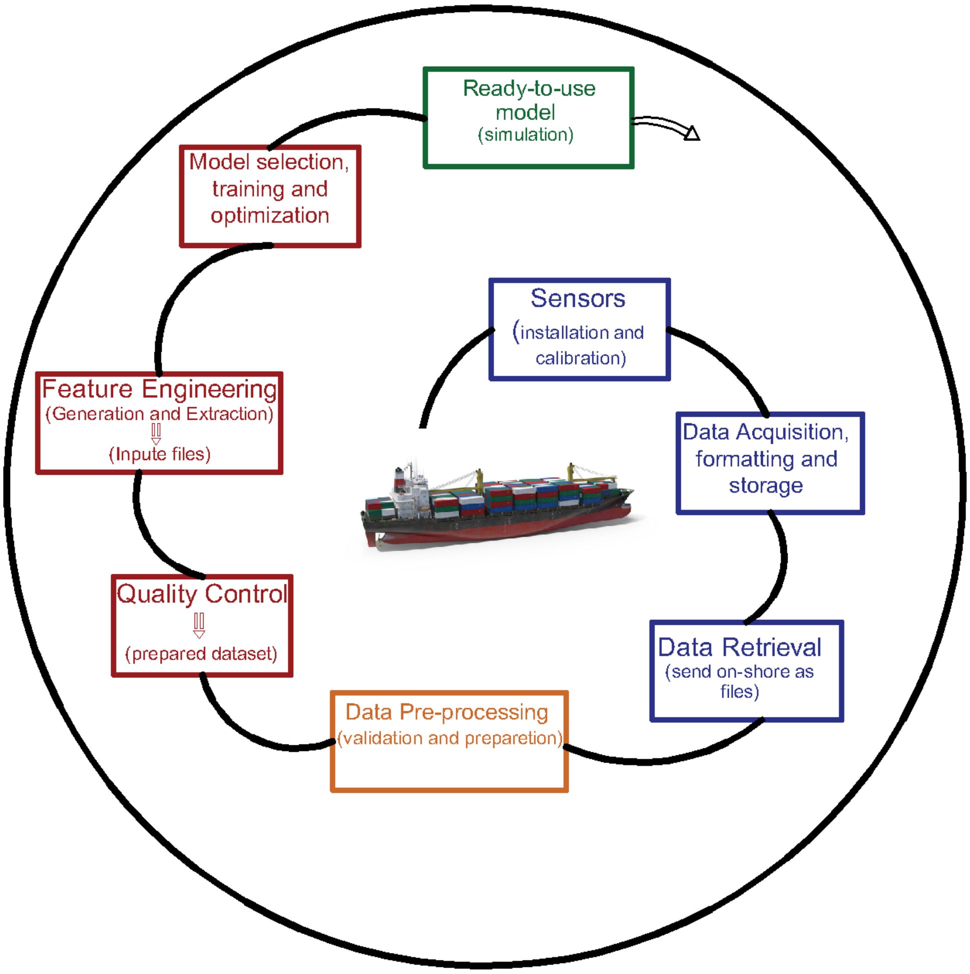


Figure 1.3 Schematic representation of the followed procedure for the data-driven ship propulsion modeling.

2 Data Acquisition

For the present study a large set of data acquired from the operation of Container ship, was made available thanks to the eagerness of the people that are managing the LAROS platform by Prisma Electronics. The ship's main particulars are presented in Table 2.1.

Table 2.1 The main particulars of the ship that provided the operational data for this study.

Main Particulars	
Shit type:	Container ship (2550 TEU)
Length Between Perpendiculars	199 m
Breadth (moulded)	30.2 m
Depth (moulded)	16.7 m
T_{MAX} / $T_{BALLAST}$	11.5 m / 6 m
Engine's NCR	19,404 kW
Engine's rpm (NCR)	96

The aforementioned container ship was equipped with sensors that were monitoring the propulsion parameters of interest, as well as the loading condition of the ship and the relative wind speed and direction. Specifically, all the available parameters from the ship's operation are presented in Table 2.2.

The parameters of Table 2.2 were continuously sampled during the ship's operation with a sampling period of 1 minute. For every recorded value the corresponding timestamp was stored as well, in the following format: *MM/DD/YYYY HH: MM: SS*. Data collected over a period of almost 19 months, from December 2016 till May 2018, were provided in the form of csv files. The recordings actually cover the 69.52% (531,560 recordings in 764,637 minutes period) of the total time, which could be considered as the time that the ship spent sailing, and so the port time was about ~30% .

For the recording of the parameters' signals a variety of sensors had to be installed and calibrated. Then the established by LAROS remote monitoring system, is responsible for the automatic and wireless data collection and their synchronization, before storing them to the ship's server and sending them to shore-based computers.

More specific, the LAROS Continuous Monitoring System relies on collector devices. These are connected to the analog or digital signals of sensors and instruments of the vessel. They offer a remotely controllable sampling rate and they analyze the retrieved signal and calculate the required parameters. Also, they set up a high-quality wireless network (with protocols based on IEEE protocols and ZigBee compliant) inside the vessel to transmit the data to the gateway. Through the gateway, all the measured and processed parameters are stored in the SQL database of LAROS Server. Afterward, the onboard server periodically produces binary files and compresses them in order to transmit them via FTP (File Transfer Protocol) or e-mail to the designated data center, with a transmission period which can be set down even to 5 minutes, (Themelis, et al., 2018a).

From that point and onwards, the data are ready to be utilized in any modeling or performance analysis task. However, a stage of pre-processing is proved to be very valuable (see Chapters 3 and 5).

Table 2.2 The recorded parameters that are included in the available dataset.

Label	Units
Speed over ground (SOG)	Knots (kn)
Speed through water (STW)	Knots
Draft mean	Meters (m)
Trim	Meters
Rudder Angle	Degrees (deg.)
Propeller shaft rpm	Revolutions per minute (rpm)
Propeller shaft power	Kilo Watts (kW)
Propeller shaft torque	Kilo Newtons * Meters (kN*m)
Main Engine start air pressure	Bars (bar)
Main Engine’s Fuel Oil Consumption (FOC)	Tonnes per 24 hours (tn/24hr)
Wind Speed	Meters per second (m/s)
Wind Direction	Degrees
Ship's Heading	Degrees
Wave height	Meters
Longitude	Degrees
Latitude	Degrees

Even though the present study focuses on the pre-processing of the data and on the propulsion modeling, as marine engineers and for integrity reasons, it is useful to discuss briefly the different types of sensors that are required for recording the signal of the parameters of interest.

Firstly, the various sensors that are used for recording the parameters of Table 2.2 are mapped and grouped in Table 2.3, and afterwards their role is analyzed.

Table 2.3 The measuring devices that are required onboard the ship to monitor the parameters of interest.

Measuring Device	Measured Parameter
GPS	SOG, Longitude, Latitude
Pressure sensor	Draft, Trim
Speed Log	STW
Shaft torque meter	Propeller shaft torque, rpm and (calculated) power.
Mass flow meter	Main Engine’s FOC
Anemometer	Wind Speed, Wind Direction
Rudder angle indicator	Rudder Angle
Gyrocompass	Ship’s Heading

i. GPS (Global Positioning System)

The GPS retrieves information about the ship’s position in global coordinates (longitude, latitude) and from the arithmetical derivation of the ship’s position, the ship’s Speed Over Ground (SOG) is obtained. The GPS operation requires constant communication with a system of satellites in order to locate the ship’s position and usually has an accuracy of a few meters.

ii. Pressure sensor

The draft of the ship can be estimated by the hydrostatic pressure on the hulls bottom surface. Sensors that measure the pressure are placed on the outer surface of hull’s bottom and can deduce the instantaneous draft of the hull at the position that they are installed. From the measurement of the draft on two different longitudinal positions of the hull, the ship’s trim can be calculated.

iii. Speed logs

For measuring the ship’s Speed Through Water (STW), two popular type of sensors exist.

a) Doppler log: An acoustic speed log based on the Doppler effect in which the wave lengths of moving objects appear to shift in relation to the observer. This shift can be converted to speed, thereby giving a very accurate result. The Dual Axis Doppler Speed Log utilizes the Doppler shifted returns from high frequency acoustic energy transmitted into water to provide precise speed data, distance travelled, and water depth below the transducer. The transmitted signal is scattered back from the sea bottom and/or scatters in the water mass. The system amplifies the received signals and processes them to determine the Doppler shift.

b) Electromagnetic log: The electromagnetic log works by generating a small alternating current in a transducer producing an electromagnetic field in the adjacent water. As the vessel moves through the water, the voltage proportional to the speed is generated at 90 deg to the direction of travel. This signal voltage is detected by the probes and transmitted to the master electronic unit where it is amplified and processed digitally before being passed to the speed and distance displays.

iv. Shaft torque meter

The shaft torque meter is a piece of equipment the measures the torque and the rotational speed of the shaft, and multiplies them to estimate the transmitted power’s value. The instrument consists of strain gauges, arranged on a ring and mounted directly on the shaft for the continuous monitoring and logging the aforementioned values. The basic principal of operation is that any deformations of the strain gauges are transferred into voltages deviation which determine the strain of the shaft.

v. Mass flow meter

When it comes to measuring the fuel consumption in a ship, the most reliable way is to do it via mass flow meters because they eliminate the need for converting the volumetric flow into a mass flow, according to the fuel’s density estimations. They are also known as Coriolis mass flow meters. The reason is that the Coriolis acceleration induces oscillations to the tubes of the device, that depends on the mass flow in them. As a result, the magnitude and the frequency of these oscillations help determine the fuel mass flow through the tubes.

vi. Anemometer

The wind anemometer is a device that provides both, the relative speed and direction of the wind with respect to the ship's orientation. It consists of a helicoid propeller and a vane the measure the wind's speed and direction, respectively. The angular displacement of the vane helps estimate the wind's relative direction, while the rotational speed of the helicoid propeller helps estimate the wind speed.

vii. Rudder angle indicator

The rudder angle indicator is an electrical device that measures the actual angle of the rudder. It consists of the two parts, the transmitter which is mounted on the steering system of the ship (steering gear room) and the receiver which is placed in the wheelhouse and displays the transmitters signal. The measuring accuracy is usually below the range of ±0.5° at common angles and ±1.5° at hard over rudder.

viii. Gyrocompass

The gyrocompass is a form of gyroscope (non-magnetic compass) that is used in ships for monitoring their heading orientation. It is based on a fast-spinning disc and the rotation of the Earth, to find geographical direction automatically. It has the ability to point always the true north, and so the ship's heading is accurately estimated with respect to this direction.

3 Data pre-processing

The present chapter aims to delve deeper into the nature of the ship's data by visualizing and analyzing them. Even though acquiring a set of data from the operation of a ship is a meaningful endeavor from scientific, technological and business perspective, the real value of these data will not arise without an effective pre-processing. The reason is that the collected dataset should prove to be a reliable and well-aligned realistic approximation of the occurring situations that a ship is facing. Considering that, the first plausible step would be to inspect visually the collected data, since this is a common quality control measure in many engineering activities and the human brain is still better in critical thinking than any known algorithm. As a next step, the data should be contrasted with the theory and then an effort to correct or clean the dataset from anomalies and outliers could help us improve their intrinsic information. However, after any interference in the original dataset, a quality control phase should follow in order to investigate the resulting effects on it.

3.1 Data Visualization

i. Plotting in the time domain

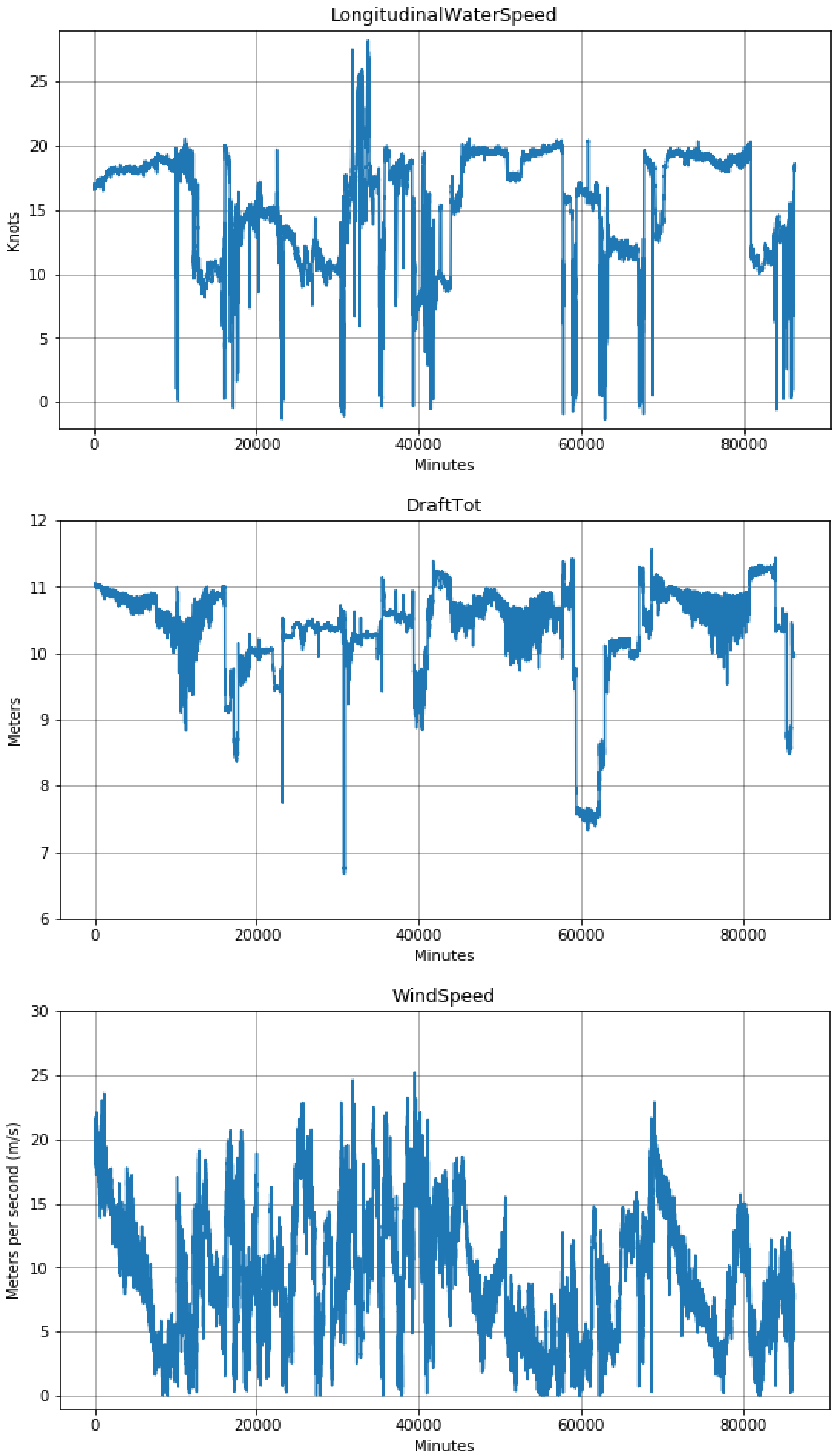
Data collected from a ship are always time-dependent and each data point has its unique timestamp as an ID and when the collected data are synchronized all the parameters share the same timestamps. As referred in Chapter 2, our data set originally consists of 16 parameters stored digitally over a period of 18 months. All these parameters should be plotted over time and visually checked but before that, the time scale of the graphs should be considered.

On the one hand, plotting over short periods of time (few hours) will present recorded signals in great detail but this would be just a small, and not representative, fraction of the data. On the other hand, plotting over large periods of times (few months) can provided useful information concerning the operational profile of the ship and helps to identify some patterns. But an important amount of details is missed due to the scale of the graph. For these reasons it was decided to plot over three different time scales:

i. over 1-day,
ii. over 1-week,
iii. over 2-months.

These characteristic plots are presented in Figures 3.1 to 3.3.

Furthermore, when plotting over short or medium size time periods, it is important to do it for specific sailing scenarios. For example, open-sea/steady sail, coastal navigation or when approaching ports.


LongitudinalWaterSpeed

DraftTot

WindSpeed

Figure 3.1 Plot over time for three measured parameters, on a 2-months period of ship's operation.

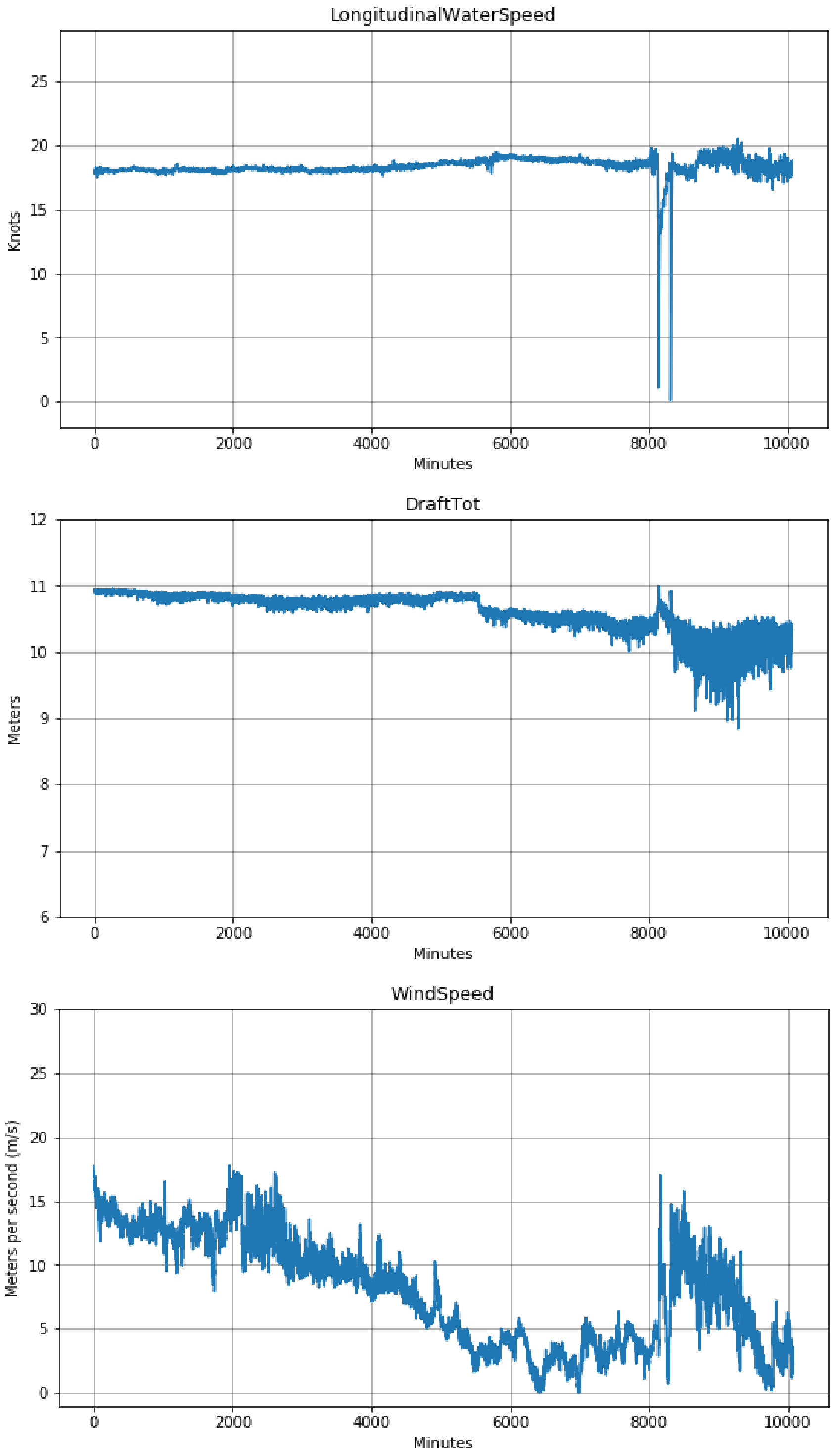


Figure 3.2 Plot over time for three measured parameters, on a 1-week period of ship's operation.

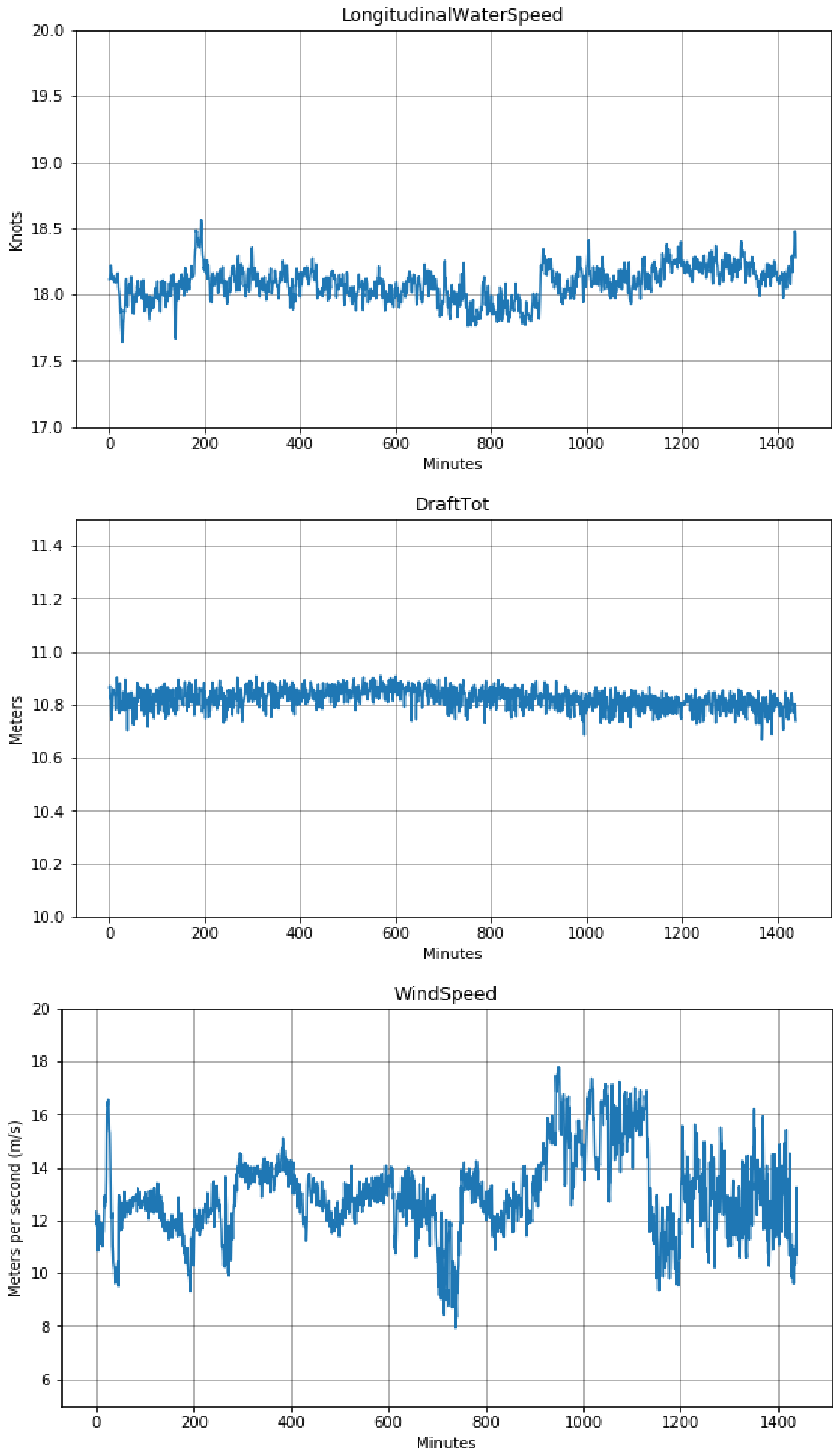


Figure 3.3 Plot over time for 3 measured parameters, on a 1-day period of ship's operation.

We choose to demonstrate three specific parameters in the discussion of plotting over time because each one of them represents one of the three main categories of ship performance monitoring parameters. As an operational parameter, we have the ship's *Longitudinal Speed Through Water* (STW), as a loading condition parameter we have the *Draft* of the ship and as an environmental parameter, we have the *Wind Speed*. Also, these 3 different families of parameters tend to have different statistical behaviors in relation with time. Operational parameters vary over short periods of time but follow some kind of patterns like: rapid increase, relatively steady (between some intervals), rapid decrease. Loading condition parameters are quite steady while the ship is sailing and then instantaneously change value when the ship loads or unloads cargo at the port. Environmental parameters always seem governed by randomness because they are characterized by spatial and temporal variation and because the ship is also moving through space, over time. The recorded signal of their values is really complex and it is difficult to spot any regularities.

The procedure of inspecting this kind of plots and trying to explain them can be supported by plotting alongside them a map projection, of the ship's path. An example of this kind of plot is given in Figure 3.4. In this way while examining the signal of some measured parameter we also have information regarding where the ship was sailing in that specific moment. For someone with relative experience on ships' operations it is useful to know if what seems like an anomaly on a recorded signal happened during open-sea steady sail, coastal navigation or when approaching a port.

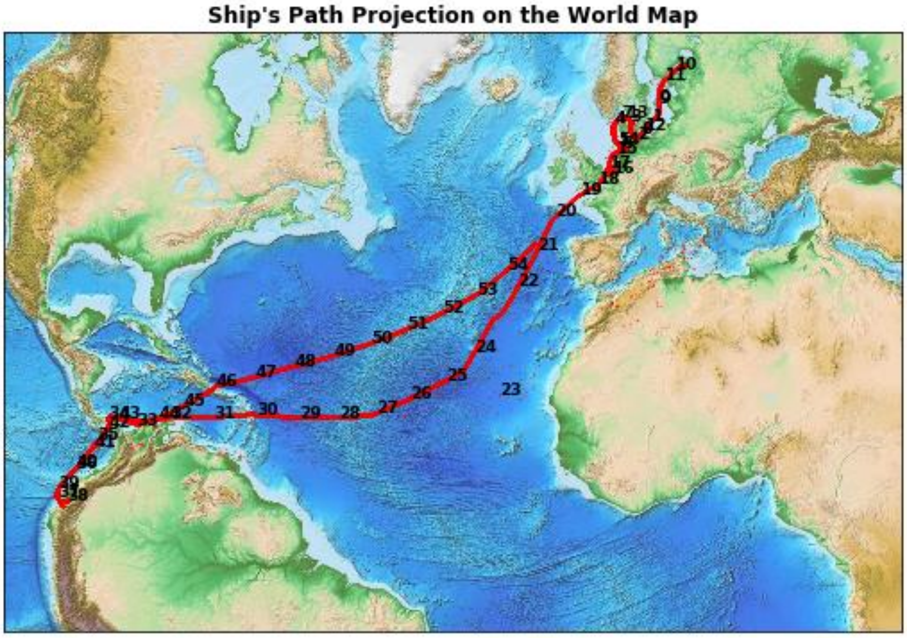


Figure 3.4 Ship's path on a world map projection with (relative) timestamps on. Each number on the map refers to thousand minutes.

When specific data points of a recorded signal are investigated for anomalies it is useful to plot a short time period around the examined timestamp and print also the timestamps along the ship's path on the map. For simplification, we print in units of thousand minutes.

Other conclusions that could be drawn from plotting the data over time, are related with the existence of noise or outliers, and the necessity for filtering or smoothing the data. In Figure 3.4 the existence of noise and outliers on the GPS signal is obvious. For instance, the data point that corresponds to the timestamp 23 is severely drifted from the ship's path. This issue however, is addressed in detail on a following section.

ii. Parametric plots

After inspecting the signals of the measured data over time, the next step should be to investigate the correlation between these parameters from the recorded data. In this process it is important to have a prior estimation of the expected correlation between these parameters, based on the physical laws that govern the phenomena in which they are involved. There are certain equations that involve some of the parameters for which we are particularly interested, in a simple and straight forward way. Those are:

- For the engine's operation:

$$P_{en} = \mathrm{Q} \cdot 2\pi \cdot n \qquad (3.1)$$

  where P_{en} is the power outcome of the engine, also known as Break Horse Power (BHP), Q is the torque on the crankshaft and *n* are the revolutions per second (for S.I. units) of the engine.

- The empirical Propeller Law:

$$\mathrm{P}_{Prop} = \mathrm{c} \cdot V^3 \qquad (3.2)$$

  where P_{Prop} is the power consumed by the propeller, *c* is a constant depending on the specific hull and propeller design and the specific loading and weather conditions, *V* is the speed of the ship. The propeller law however, can be written and in another form:

$$\mathrm{P}_{Prop} = \mathrm{c} \cdot n^3 \qquad (3.3)$$

  where now *n* are the propeller revolutions, which are proportional or identical (when there is no reduction gear) to the engine's revolutions. From the two above equations we conclude that the speed *V* is proportional to the propeller's revolutions *n* and from (3.1) we get that Q is proportional to V^2.

- The Calm Water Resistance Coefficient:

$$\mathrm{C_T} = \frac{\mathrm{R}}{\frac{1}{2}\rho \mathrm{SV}^2} \qquad (3.4)$$

  where *R* is the measured resistance force, *ρ* is the fluid's density, *S* is the wetted surface. The water resistance is the force that needs to be overcome in calm sea, by the propeller's thrust, in order to sail at the desired speed V, and this is achieved when the propeller's effective power (P_{eff}) is equal to:

$$\mathrm{P}_{eff} = V \cdot R\,. \qquad (3.5)$$

The wetted surface S, which depends on the draft of the ship, is influencing the hull's resistance. However, the draft of a ship is relatively constant during a voyage and so it is considered uncorrelated with the power and the speed because it is actually an operational decision at which speed to sail when then draft is already known.

Last but not least, there is no need to involve equations to acknowledge that the Fuel Oil Consumption (FOC) is monotonously increasing with the engine's power and RPM.

Table 3.1 The expected correlation (based on the theory) between the available parameters.

EXPECTED CORELLATION TABLE

0: not correlated, **1**: loosely correlated, **2**: strongly correlated

	Speed Over Ground	Longitudinal Water Speed	Draft Total	Trim	Rudder Angle	Propeller shaft RPM	Propeller shaft Power	Propeller shaft Torque	M/E FOC	M/E Start Air Press	Wind Speed	Wind Direction	Heading	Longitude	Latitude
Speed Over Ground	╳	2	0	0	1	2	2	1	2	1	0	1	0	0	0
Longitudinal Water Speed		╳	0	0	1	2	2	1	2	1	0	1	0	0	0
Draft Total			╳	0	0	0	0	0	1	0	0	0	0	0	0
Trim				╳	0	0	0	0	1	0	0	0	0	0	0
Rudder Angle					╳	0	0	0	1	0	0	0	0	0	0
Propeller shaft RPM						╳	2	2	2	1	0	0	0	0	0
Propeller shaft Power							╳	2	2	1	0	0	0	0	0
Propeller shaft Torque								╳	2	1	0	0	0	0	0
M/E FOC									╳	0	0	0	0	0	0
M/E Start Air Press										╳	0	0	0	0	0
Wind Speed											╳	0	0	0	0
Wind Direction												╳	0	0	0
Heading													╳	0	0
Longitude														╳	0

In Table 3.1 the “expected” or “estimated” correlation between each of the available parameters is presented. Three different classes are used to express the expected degree of correlation among a pair of parameters. They could be characterized either uncorrelated or loosely correlated or highly correlated. At this point of the analysis, we should state again that only the physical correlation between the parameters is considered and not the operational, that many times exists, but varies among different type of ships and/or operators (i.e. draft and trim that are carefully chosen or draft and speed, may be operationally correlated).

Afterwards, we plot the dataset’s parameters against each other (Figures 3.5 to 3.11) in order to confirm or reject our prior beliefs. In this process, calculating the linear correlation coefficient for any pair of parameters offers some quantitative indications on whether this pair is correlated or not, to the degree we thought it is. The Table 3.2 shows the linear correlation coefficient (Pearson product-moment) between the parameters of each figure.

Table 3.2 The calculated value of the linear correlation coefficient for seven cases from the discussed parameters

Figure No	Correlation Coefficient Value	Expected
3.5 (STW-Propeller rpm)	0.956	strongly correlated
3.6 (Propeller rpm- FOC)	0.973	strongly correlated
3.7 (Draft- FOC)	0.125	uncorrelated
3.8 (Draft- Trim)	0.473	uncorrelated
3.9 (STW- Wind speed)	-0.043	uncorrelated
3.10 (Shaft power- ME Start. Air Press.)	-0.010	loosely correlated

For Figures 3.5 and 3.6, the parameters are declared strongly correlated and this is confirmed by the value of the correlation coefficient. In Figure 3.9 is also confirmed that there is no statistical correlation between the speed of the ship and the wind speed. In contrast, it is noticeable that for Figures 3.7 and 3.8, whose parameters are declared uncorrelated, the linear correlation coefficient is far from zero. This is due to an operational pattern of the ship. No physical law states that the trim of the ship is somehow dependent with the draft but in this particular case the trim increases with the draft, as the data prove. In general, the trim-draft relationship is defined by the designers. In contrast, the draft seems correlated with the fuel oil consumption only because the ships do not reduce their speed when they are more loaded and consecutively have larger wetted surface area and as shown by Equation (3.4) larger water resistance that requires more fuel consumption to achieve the same speed. In Figure 3.10 the correlation value, also does not come in agreement with our estimation but it can be assumed that a non-linear correlation exists by the fact that when the engine’s power is low, the starting air pressure becomes lower too.

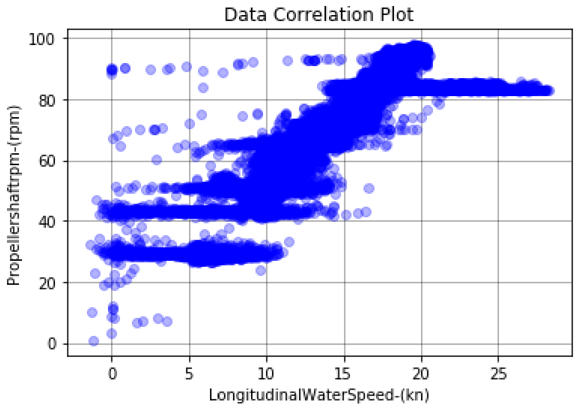


Figure 3.5 Parametric plot of STW vs Propeller Shaft RPM.

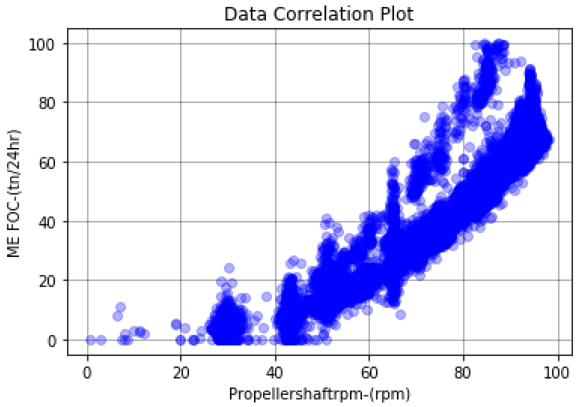


Figure 3.6 Parametric plot of Propeller Shaft RPM vs M/E FOC

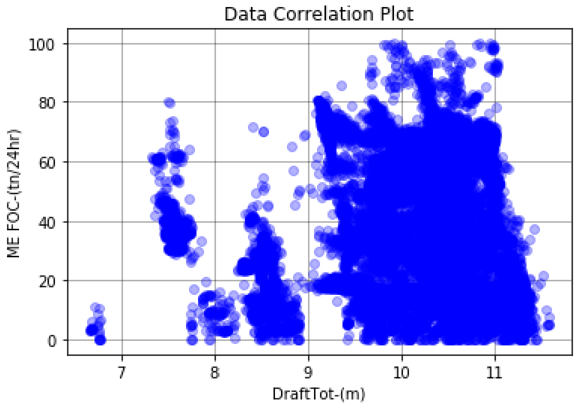


Figure 3.7 Parametric plot of Draft vs M/E FOC

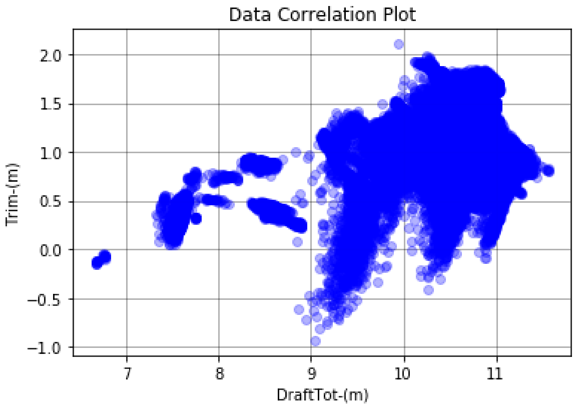


Figure 3.8 Parametric plot of Draft vs Trim

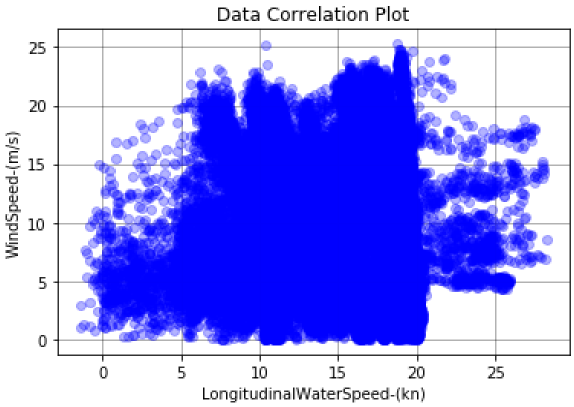


Figure 3.9 Parametric plot of STW vs Wind Speed

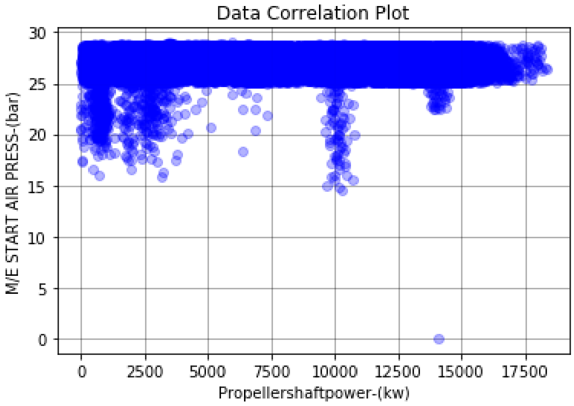


Figure 3.10 Parametric plot of Propeller Shaft Power vs M/E Start Air Pres.

iii. 3-Dimensional Plots

In the previous paragraphs, we searched for correlations between the propulsion parameters and classified them as physically or operationally correlated, through the use of 2-D parametric plots. The problem of evaluating a ship's performance however, is multi-dimensional and it involves many interdependent parameters. An effort to visualize the data to more than two dimensions has a lot to offer in the discussion. It helps to identify possible patterns in the operation of the ship and demonstrates the way that some parameters interact with each other. In addition, plotting in three dimensions is one more convenient way to make a reality check between data and theory.

For a data set that includes 16 different measured parameters, it is not clear which three of them should be plotted against each other in order to create a meaningful scatter plot that will provide us with improved insight on the acquired data. Figure 3.11 shows a Speed-Draft-Trim scatter plot where the data are filtered to include only calm sea condition and transatlantic trips. The color scale is proportional to the speed-axis vertical) for extra clarity. We notice that for the whole range of ship's draft (8.5 m to 11.5 m), almost the same service speed is chosen. Also, larger trims (from 1 m to more than 1.5 m) appear when the speed approaches its maximum values and when the trim is around zero the speed is significantly reduced.

In a 3-D scatter plot a fourth dimension could be presented if the color scale is set in proportion to a variable that is not plotted along any of the x, y, z axes. In Figure 3.12 these three variables are plotted but this time the data are not filtered for calm sea conditions and the wind speed value is set to be proportional to the color scale. The datapoints on the speed-power plane take the familiar form of a speed-power curve that follows the 'propeller law' of equation (3.2). As it was expected, higher values of the power, for the same value of speed, are observed when the wind is stronger.

It should be mentioned again that these plots are useful only for validation of the data and of our understanding on the problem. Even if we create a 4-D plot, where the color scale is analogous to the point in time that each specific datapoint was recorded, we would be very fortunate if we could distinguish any losses in the ship's performance. And even then, many more parameters should be taken under consideration. Hence, there is no point in evaluating the vessels performance simply by graphical trends of the data because it is actually a more complex problem involving many more parameters.

As an alternative to the 4-D color scale plot, we can have information from 4 dimensions plotted on the 3-D scatter plots in the form of parametric plots. For example, in Figure 3.13 the x, y and z axes of the 3-D plot represent the Wind Speed, the Draft and the Power, and an identifying color is assigned to each discrete value of the selected parameter, Speed. For this purpose, the data are filtered around the desired values of the parameter, in our case 16±0.1, 17±0.1, 18±0.1 and 19±0.1 knots. Again, the power demand increases for the higher speeds and when sailing at any particular speed, the increase of wind speed causes the power demand to increase. The observed trend is well in accordance with the equation that estimates wind resistance, which is identical to equation (3.4), with V being the relative speed.

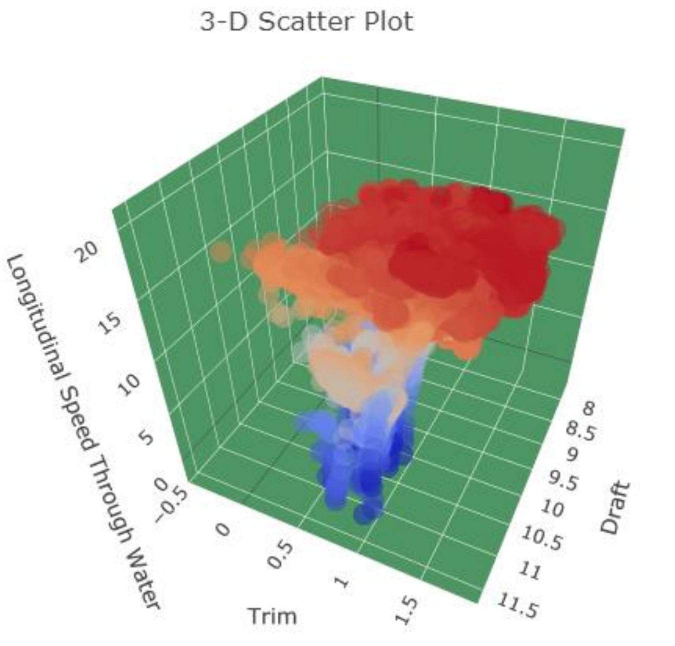


Figure 3.11 Three-dimensional scatter plot of ship's Speed vs Draft vs Trim. The color scale is set proportional to the ship's speed for clarity.

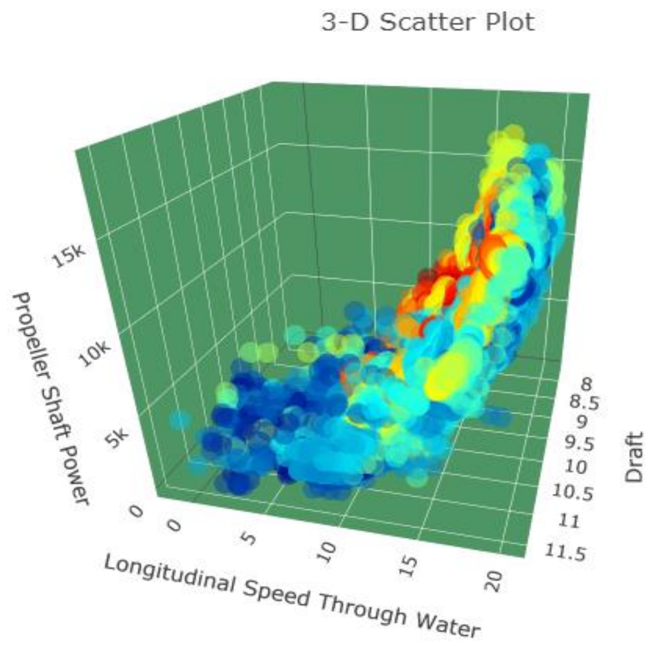


Figure 3.12 Three-dimensional scatter plot of Power vs Speed vs Draft. The color scale escribes the Wind Speed, with range from 0 to 25m/s (blue to red).

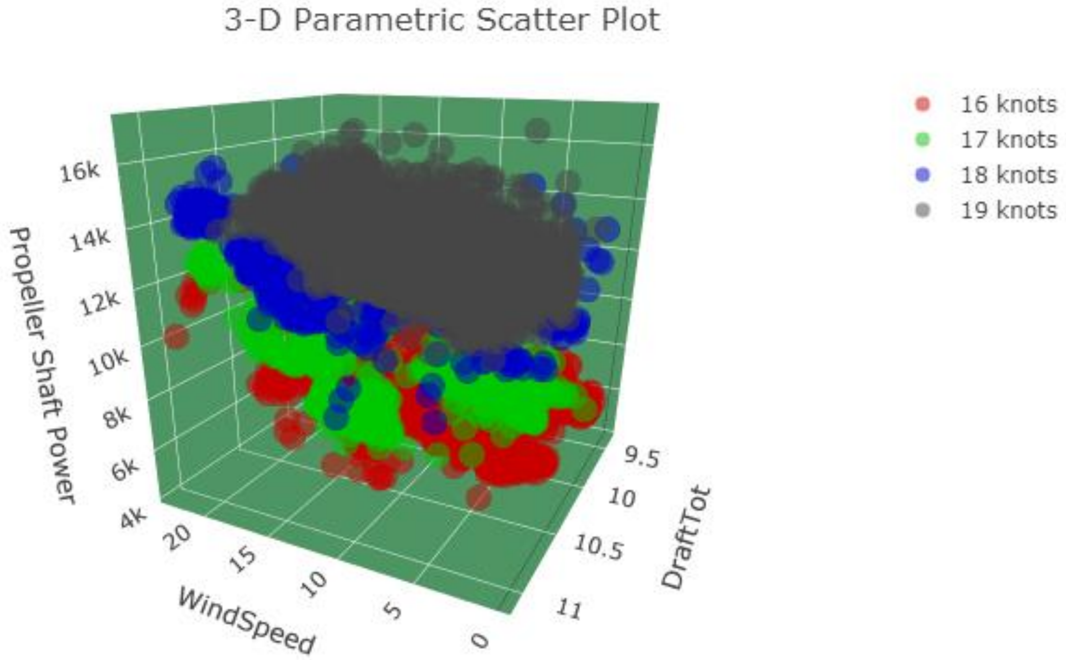


Figure 3.13 Three-dimensional scatter plot of Power vs Draft vs Wind Speed with parameter the ship's speed (STW). Each color corresponds to a specific speed, according to the legend (top-right).

3.2 Data Correction

The main target of the present section is to present an algorithm created for spotting and correcting suspicious measurements on time-dependent data that are related to some kinds of physical phenomena. It is intended to make a modest correction on data like a liquid's flow rate, a body's movement in space, the revolutions of an engine or the angular position of a rudder. The basic idea behind this correcting algorithm is that physical objects, meaning objects with mass of course, always display inertial behavior in the sense that during an adequately short time window in the future, their past value or state is not expected to change excessively, because inertia means resistance to any change in current motion or state. This correcting algorithm is presented theoretically and additionally several examples of its application on different parameters of the dataset are given.

First of all, the necessity for such an algorithm should be pointed out. In the previous section, the whole dataset was visualized in various ways. It was quite clear in some of these plots that on the existing data, many datapoints are included that could possibly be outliers, noise or anomalies. Actions have to be taken in order to exam which of these data points, that seem faulty to the naked eye, should actually be corrected or discarded. Certainly, this is not a simple task, since there are not well-established methods for "cleaning" or correcting this particular kind of interdepended and time-depended data. Careful consideration on the nature of the tested parameter should be taken before applying any such method or algorithm. However, the ability to check the implications of any such algorithm, applied on the data, in a straight forward and systematic way is 'allowing' us to attempt some new approaches on the subject.

The most profound case of outliers' existence is spotted on the *Latitude* parameter of the GPS signal. This parameter, along with the *Longitude*, are describing the ship's course during the trips. Surprisingly, the number of outliers in the latitude's signal was much larger than in longitude's (see Figure 3.23). In Figure 3.14 the value of the latitude coordinate, for a 6-month period, is plotted against time and the points that break the continuity of the ship's path are the ones that we visually identify as outliers and would like to correct. In figure 3.15 the latitude and longitude signals are plotted on a map projection and it becomes clear that these points cannot be true measurements that describe the location of our ship. They seem to form a similar, to the ship's original path, pattern. A fact that declares some kind of drifting in the recording or transmission of the GPS signal, on frequent intervals.

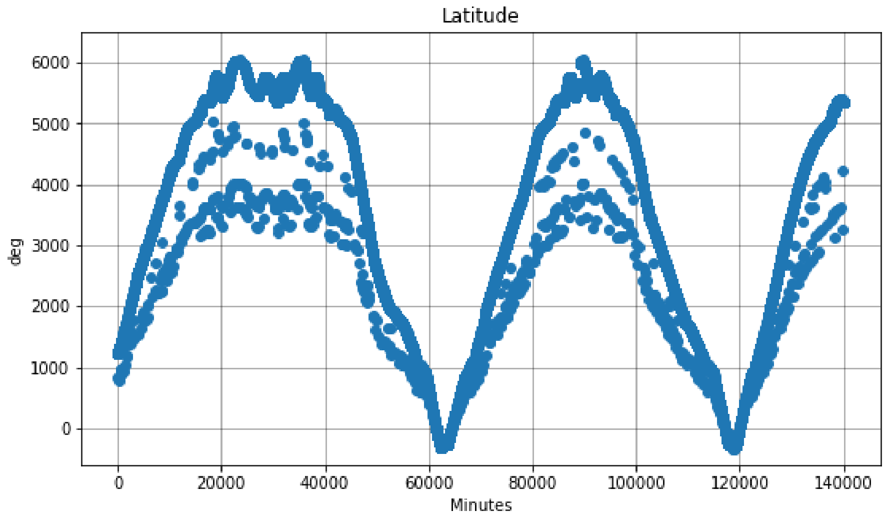


Figure 3.14 The latitude signal as acquired from the GPS on-board the ship. Raw-data time plot.

In Figures 3.16 the corrected latitude signal is plotted on a time-plot and in Figure 3.17 the corrected ship path is projected on the map. No more points seem to break the continuity of the ship's path now and the whole amount of information regarding ship's position seems to be preserved. The way that this was achieved is explained in the next paragraph.

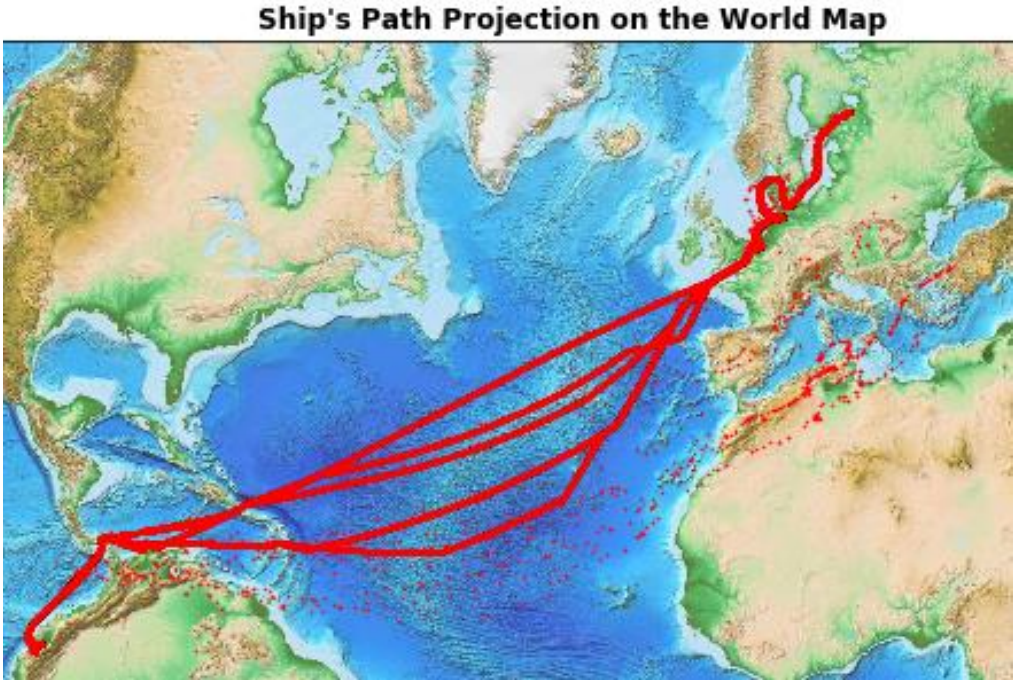


Figure 3.15 The latitude and longitude coordinates as acquired from the GPS on-board the ship. Raw-data map projection.

In order to characterize a data point in the latitude coordinate's signal as an outlier and correct it, we should consider the dynamics of the ship's movement through space. The latitude and longitude coordinates correspond to unique positions on the earth's surface, on the same way that x and y coordinates do on a cartesian plane. Therefor the latitude can be thought as the y coordinate of the ship's position. In this case, speed over ground and the derivative of the ship's heading are the parameters that determine the rate of change of the ship's coordinates. If the speed and heading are constant, the rate of change is constant and for constant sampling rate, the absolute difference (distance) of consecutive latitude (and longitude) data points is also constant. When real ship data are processed, we cannot expect to encounter absolutely steady conditions. However, any acceleration or deceleration that occurs to the ship is not expected to cause a displacement, much larger than the previous, to the next data point. That is due to the magnitude of a large ship's inertia and on condition that the sampling frequency is efficiently high, in comparison with the dynamic of the observed phenomenon. If we express it in calculus terms, the derivative of the signal with respect to time should not rise instantaneously to any arbitrary value.

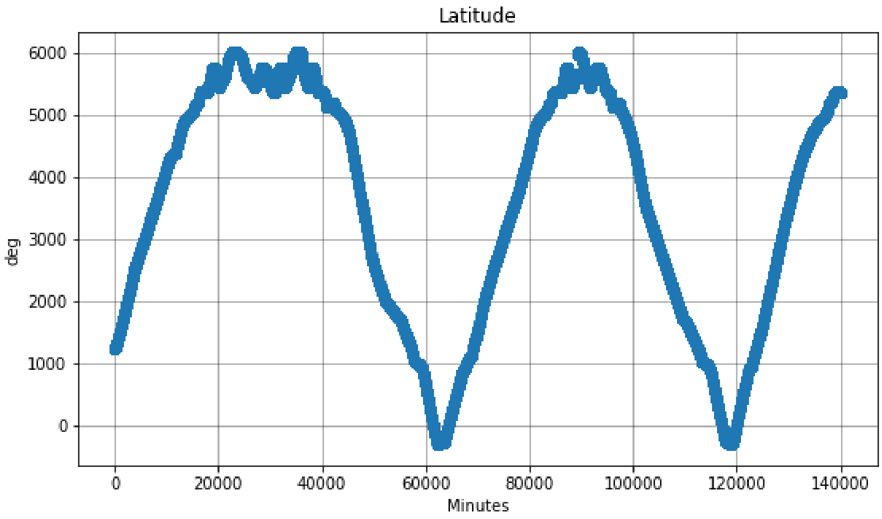


Figure 3.16 The latitude signal plotted over time, after the correction of outlying values.

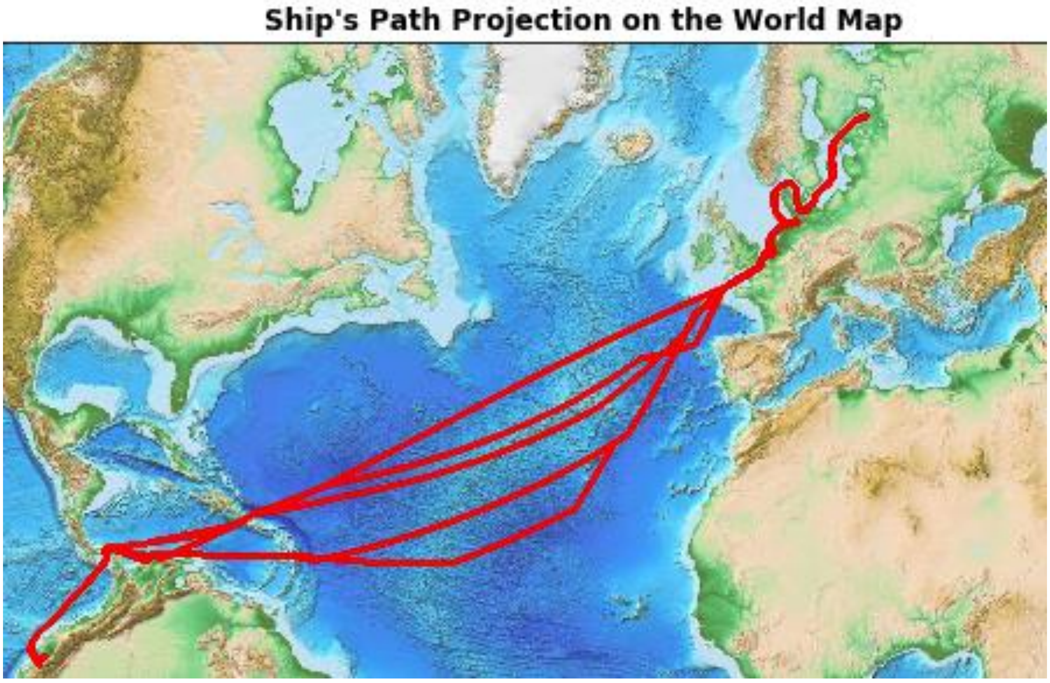


Figure 3.17 The GPS signal projected on the map, after the correction of the problematic values.

The above idea has taken the form of an algorithm that examines if a data point is diverging excessively from the neighboring data points and "corrects" it by setting its equal to the mean value of the previous and the next data point. It is the algorithm that 'corrected' the latitude signal of Figure 3.14 resulting to the signal of Figure 3.16. The same algorithm was also applied on the longitude signal, which had fewer outliers, as can be seen in Figure 3.21.

“CORRECTING ALGORITHM”

Let d be a list of data points, of a particular parameter’s signal and t a list of the data points’ timestamps. Then $d_i,\ i = 1, \dots, n$ is the ***i***-th element of the list d and $t_i,\ i = 1, \dots, n$ is the corresponding ***i***-th timestamp.

```
i.    Calculate
          $\delta d_{i-} = \frac{|d_i - d_{i-1}|}{dt_-}$
      ,where $dt_- = t_i - t_{i-1}$ and similarly,
          $\delta d_{i+} = \frac{|d_i - d_{i+1}|}{dt_+}$ .
ii.   Calculate
          $D_i = \frac{|d_{i-1} - d_{i+1}|}{dt_D}$
      , where $dt_D = t_{i+1} - t_{i-1}$.
iii.  If $dt_D \leq t_{lim}$ then
iv.       If $[\delta d_{i-} > sf \cdot D_i]\ AND\ [\delta d_{i+} > sf \cdot D_i]$ then
v.            $d_i = \frac{d_{i-1} + d_{i+1}}{2}$
```

In step (iii), t_{lim} is a user-defined parameter that can force the algorithm to skip checking the current data point. It defines what is the maximum time-gap between to measurements, that permits the algorithm to check if a data point should be corrected. In step (iv), the sf is just a scaling factor that is selected by the user, based on his understanding of the natural quantity’s dynamical behavior and maybe some experimentation.

In the implementation of the algorithm in this study, the t_{lim} value was set to 3 minutes and the *sf* had several different values, depending on the parameter that was processed each time.

The algorithm may be applied once more on the same data points but this time checking among the ***(i-2)***-th and ***(i+2)***-th elements instead of the ***(i-1)***-th and ***(i+1)***-th. This is done because it was noticed after the first implementation of the algorithm that some obvious outliers remained in the data set. That happened because some outlying points may appear consecutively. In this case the ***(i+1)***-th element is an outlier itself and the algorithm does not identify the ***i***-th element as an outlier. By pursuing one time-step further and involving the ***(i-2)***-th and ***(i+2)***-th elements we manage to drastically reduce the probability of misinterpreting a data point for two reasons. First, if we consider the outliers to be independent and identically distributed (i.i.d), the probability of having three consecutive outliers is the cube of the probability of having a single outlier, which is by definition much less than 1. Hence the value of this probability is really small. Second, the time window involved in the check of the ***i***-th element is still small (±2 mins from the present value) in comparison to ship’s dynamics for large changes of her state.

A few more examples, from the application of the correction algorithm on some of the recorded ship parameters, are given below. In every figure, with blue dots are plotted the original data points and the red curve is the ‘corrected’ signal. The effort here is to investigate qualitatively the effect of the correction algorithm on the original data points. In Figures 3.18 and 3.19 we notice that despite the fact that the signal is pretty unstable and probably noisy, not too many data points are corrected by the algorithm and in most cases, only points that stand out of the most stable fractions of the signal are affected by the algorithm. In Figures 3.20 and 3.21 we observe macroscopically the successful identification and correction of outlying points for two different kinds of sensors and physical quantities. In conclusion, from the short time period plots we can better understand the way that the algorithm acts on many different occasions and from the longer time period plots we get a better picture of what has finally happen to our original signal. A more quantitative look on the effect of the algorithm on the original data is given in the last section of this chapter.

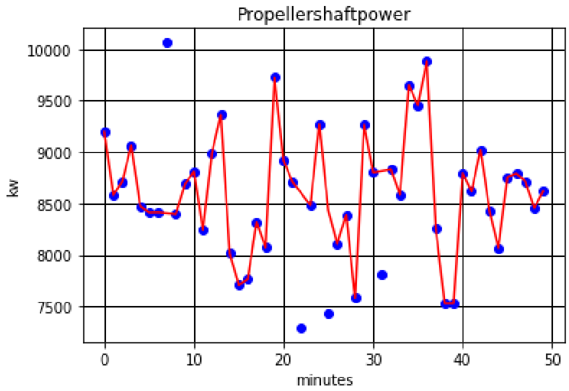


Figure 3.18 Ship's propeller shaft power measurements plotted for a period of 50 minutes. The blue dots represent the original data points and the red curve is the parameter’s signal after the correction of the data.

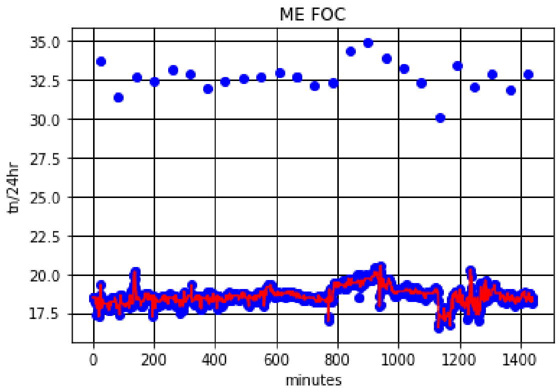


Figure 3.20 Main engine’s FOC measurements plotted for a period of 1 day. The blue dots represent the original data points and the red curve is the parameters signal after the correction of the data.

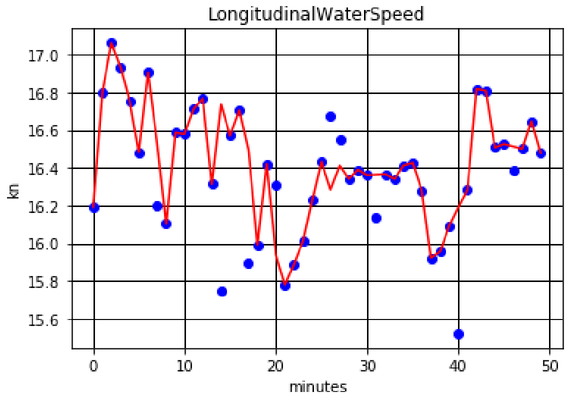


Figure 3.19 Ship's STW measurements plotted for a period of 50 minutes. The blue dots represent the original data points and the red curve is the parameters signal after the correction of the data.

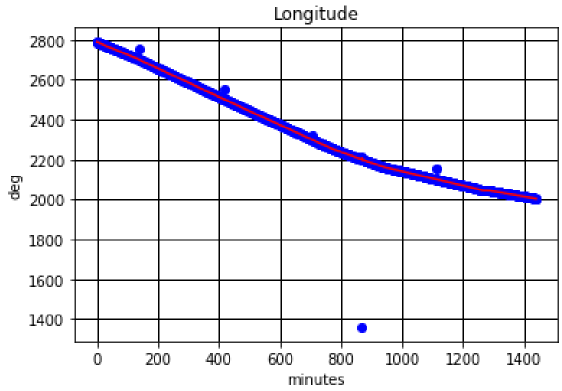


Figure 3.21 Ship’s Longitude GPS measurements plotted for a period of 1 day. The blue dots represent the original data points and the red curve is the parameters signal after the correction of the data.

3.3 Outliers Detection

In continuation of the previous section, the procedure of analyzing the data in order to identify outliers or physical impossibilities, is supplemented here. An alternative approach for identifying outliers in the data is implemented. The basic difference with the correcting algorithm of the previous section is that this method does not treat the data as time series and actually does not take under consideration the dataset's temporal information at all. The focus is given on the statistical behavior of only a few interconnected parameters. Also, no corrections are applied, hence data points identified as outliers are deleted from the data set. They are deleted because the focus is on capturing data points that fall on the tail of the probability distribution curve and which we assume that imply physical improbabilities because they are cross-checked with other parameters' values in the particular timestamp.

The idea behind the proposed method comes partially from the Chauvenet's Criterion which is the proposed method for outlier detection in (ISO 19030, 2016). Briefly, the Chauvenet's Criterion indicates that:

> The probability for the occurrence of any value d_i is computed according to the equation (3.6):
>
> $$P(d_i) = erfc\left(\frac{delta_i}{\sigma \cdot \sqrt{2}}\right) \tag{3.6}$$
>
> Where:
>
> - $P(d_i)$ is the probability of d_i
> - $delta_i = |(d_i - \mu)|$
> - $\mu = \frac{1}{N}\sum_i^N d_i$
> - $\sigma = \sqrt{\frac{1}{N}\sum_i^N delta_i^2}$
> - $erfc$ is the complementary error function
>
> A datum is considered an outlier if the inequality (3.7) is fulfilled.
>
> $$P(d_i) \cdot N < 0.5$$

However, there are two assumptions in the Chauvenet's Criterion that should be discussed.

a) The probability distribution of the data points is given by the complementary error function. This assumption actually implies that the collected data points are normally distributed in any case.

b) A frequentist notion of probability is clearly assumed on the check for the outlier, inequality (3.7). That notion requires a satisfactory large amount of data in order to provide a legit probability value and then judge upon it.

The fact that these two assumptions cannot be always realistically fulfilled, is the first reason why we had to come up with an alternative approach for statistical detection of outliers on ships data. The other reason is that we found only a few studies experimenting with the outlier detection in a ship dataset and we want avoid applying another statistical or machine learning method (for outlier detection) of general purpose.

According to ISO 19030[1] in the “Data filtering and validation” phase for “...consecutive, non-overlapping blocks spanning 10 minutes, data for every parameter shall be filtered according to Chauvenet’s Criterion.” A data block contains by definition a maximum of 40 measurements (sampling period: 15 seconds) for each parameter. In this case N ≤ 40 in the equation (3.7*)*, which may be considered a statistically significant amount of data. However, in practice, the storing frequencies of sensors’ data are not high enough for producing datasets with such a large amount of data in such a short time period. If the 10-minute period is extended the assumption of steady environmental and operational conditions is less supported. For example, if we have four recordings of the ship’s speed in a 5-minute time window it makes no sense to apply the Chauvenet’s Criterion in order to identify at least one of these four data points as an outlier. If we choose a larger time window, i.e. 30-minute, so that a sufficiently large number of speed recordings will be available, the probability of an increase or decrease in the speed due changes in the environmental conditions is quite high. Once again, outliers cannot be properly detected since the same value of speed may be an actual outlier if it occurred later in the time window, when the ship’s state was different, rather than earlier. In a hypothetical example, see in Figure 3.22 how the data point at the 8th minute is an outlier but cannot be detected because earlier in the time-window, the ships speed was lower and hence the same value (14,2 knots) is not an outlier there.

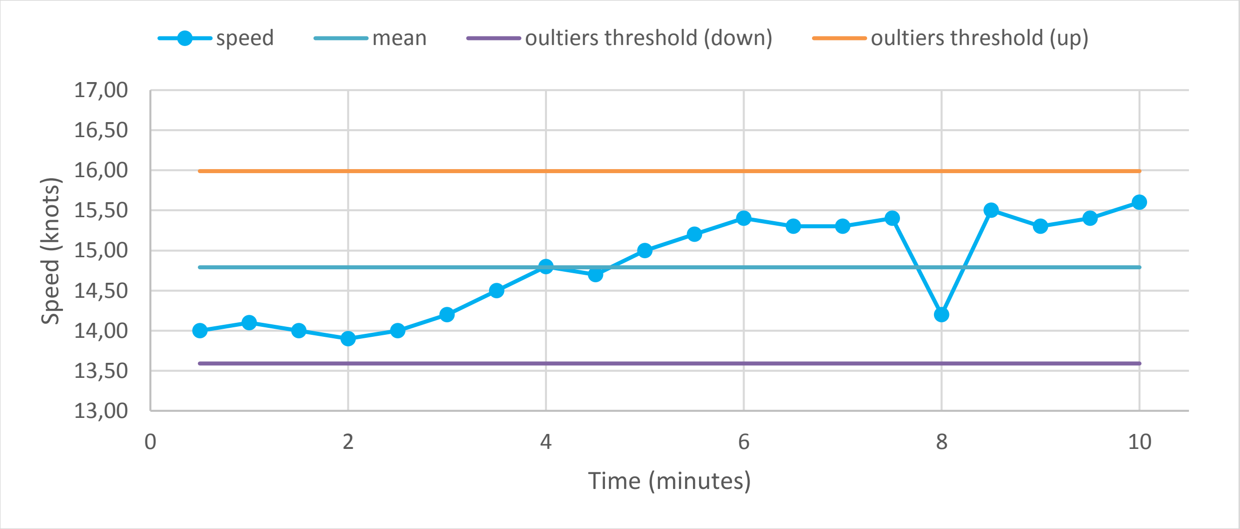


Figure 3.22 An example of a false negative in the outlier detection procedure.

In addition, data on ships are collected by different kinds of sensors and for different kinds of parameters-physical quantities. It is uncertain if the error in the measurements of every sensor is normally distributed. Also, often a signal fluctuates due to environmental factors. So, deviations in the values do not occur due to the sensors limited accuracy but is the true value of the measured quantity that oscillated during this 10-minute data block. In the absence of information from other parameters if we filter only according to the probability value given by the erfc there is high risk of biased outlier detection. Of course, it is difficult to think of a more appropriate probability density function (pdf) than the erfc but we can always avoid assuming any pdf at all.

[1] “Ships and marine technology – Measurement of changes in hull and propeller performance – Part 2: Default Method”

Finally, the procedure that we came up with, for detecting outliers in the ship's dataset, is presented step-by-step here:

- Choose a primary parameter p_1 (i.e. propeller's shaft rpm).
- Split the primary parameter in groups of values with range s (i.e. per 1 rpm).
- Group the data points according to the splits of the primary parameter, in data groups G_i. (i.e. all data points that have rpm value from 53 to 54.)
- Choose the secondary parameter p_2 (i.e. the propeller's shaft torque or M/E FOC)
- Calculate the mean value, $m_{p_{2i}}$ and the standard deviation, $\sigma_{p_{2i}}$ of the secondary parameters in each group of data G_i.
- Choose a factor k to multiply the standard deviation $\sigma_{p_2 i}$ for setting an "outlier threshold". (i.e. $k \in [2.5,3.5]$)
- If the inequality (3.8) is fulfilled, then reject the data point.

$$\left|p_{2ij} - m_{p_{2i}}\right| > k \cdot \sigma_{p_{2i}} \qquad (3.8)$$

, where p_{2ij} is the value of the j-th element in the i-th group of the p_2 parameter.

The way that the method works is presented with the help of histograms and parametric plots. In Figure 3.23 two groups of data points, with secondary parameter the propeller's shaft torque, are plotted and the "outlier threshold" is presented with the colored dot-lines for the different values of the factor k. The data group in the rpm range (87,88] includes around 7,000 observations while in the rpm range (92,93] there are around 14,000 observations. In both cases, the distribution's shape seems similar and close to a normal distribution but it is slightly asymmetric. A similar distribution appears when the secondary parameter is different. As observed in Figure 3.24 where the secondary parameter is the M/E FOC and histograms for the same groups are plotted. Because the same data groups are plotted the number of observations on each histogram remains the same as in Figure 3.23. Also, the distributions look similar but the main difference is in the value of the standard deviation, for the data group in rpm range (87,88], due to the existence of many distant outliers.

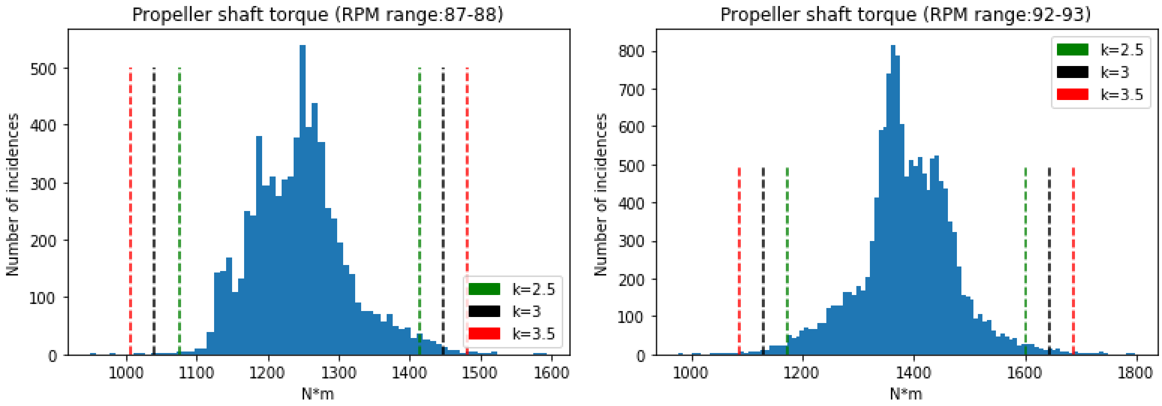


Figure 3.23 Histograms of propeller's shaft torque values for certain range of rpm values. The "outlier threshold" is plotted for different values of the factor k.

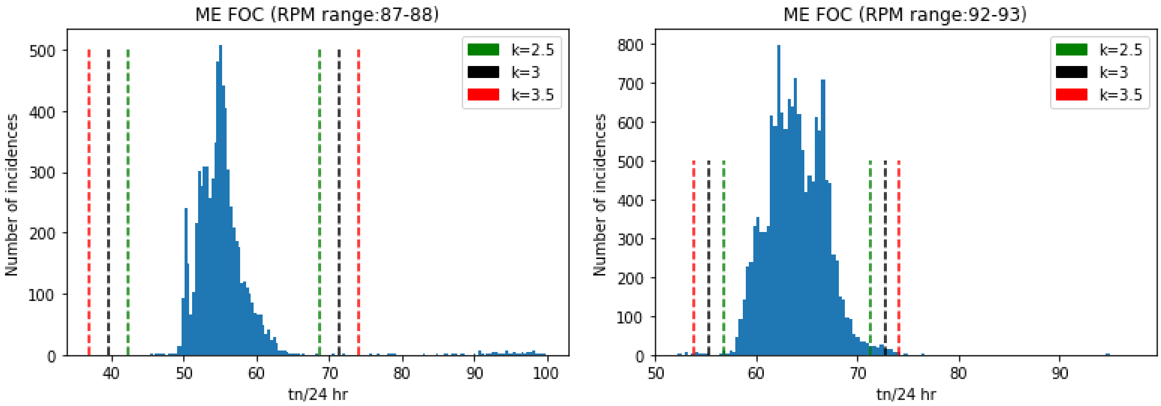


Figure 3.24 Histograms of M/E FOC values for certain range of rpm values. The "outlier threshold" is plotted for different values of the factor k.

In practice, when the above algorithm is executed the “outlier thresholds”, that are shown in Figures 3.23 and 3.24, adopt the value k=3. The procedure of outliers’ detection is completed when all data groups have been scanned. The final results of the ‘cleaned’ data set with respect to one primary (propeller shaft’s rpm) and three secondary parameters are illustrated in Figure 3.25.

An additional filtering criterion that was used at this stage is that the arithmetic value of the M/E FOC and STW should be above 3 ton/24hr and 3 knots, respectively. This is an explicit filtering criterion with no statistical reasoning behind it. Simply, we do not wish to involve in our models the cases were the ship is almost stationary.

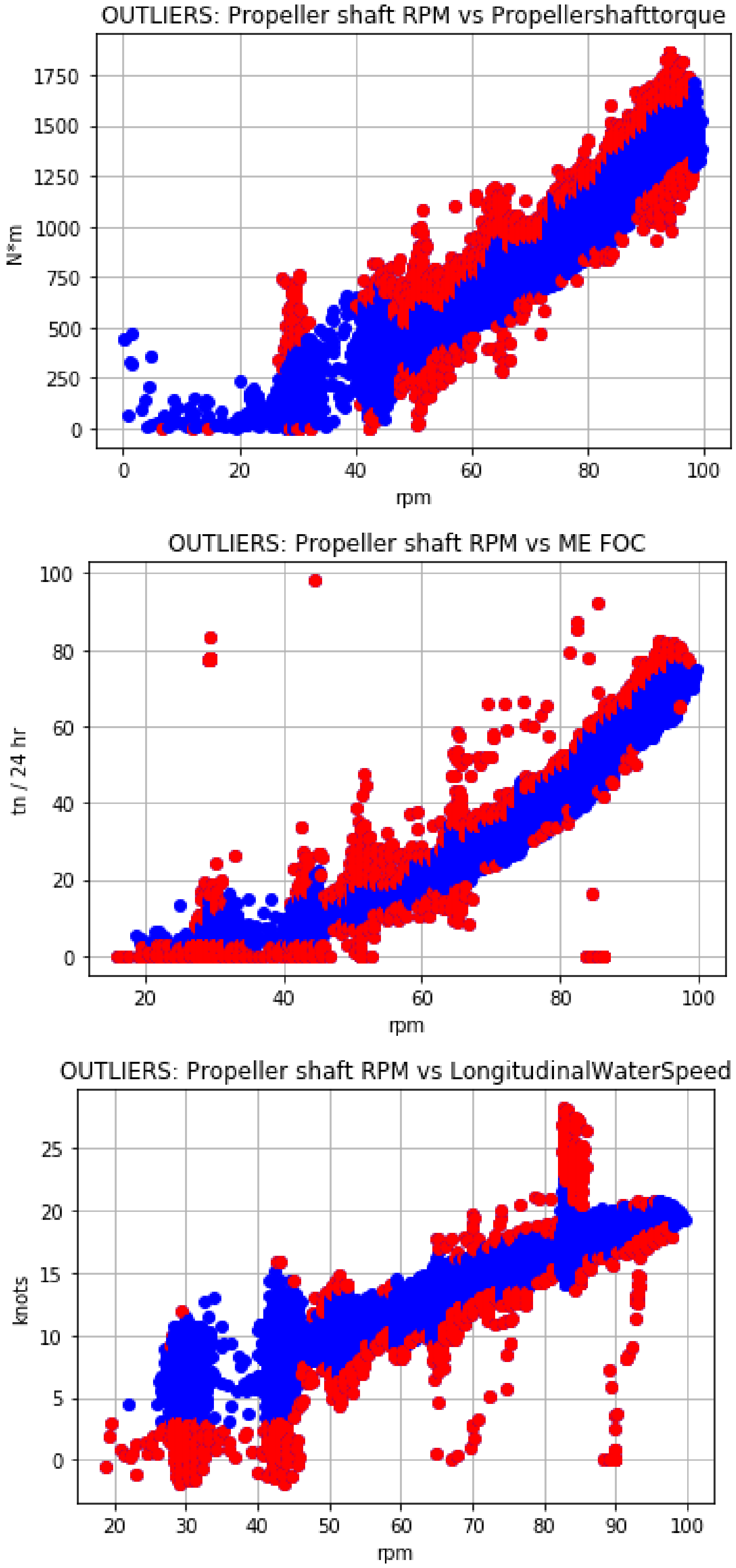


Figure 3.25 The propeller’s shaft torque, ME FOC and STW values plotted against the shaft's rpm, respectively. In red color are the data points that were identified as outliers. In the ME FOC and STW plots are also with red color (rejected) the data point.

In an alternative perspective, the described outlier detection method could be expressed in terms of conditional probabilities. For instance, we can introduce the probability of an arbitrary secondary parameter p_2 having a value larger than p_{2j}, on condition that the primary parameter p_1 lies within a thin strip $(B_l, B_u]$ (lower bound and upper bound respectively) of its sample space. Then a probability value P_0 can be selected as a threshold for outliers.

$$P\left(p_2 > p_{2j} \middle| B_l < p_1 \leq B_u\right) \leq P_0 \qquad (3.9)$$

In this way, we do not involve the sample's standard deviation explicitly and we are very close to Chauvenet's criterion formulation. However, the problem of correctly estimating the probability value remains even though we now have many more samples for estimating its value on a frequentist notion or fitting a pdf (Normal Distribution or other) with properly estimated parameters (i.e. $\bar{x}$ and σ). If we could estimate the value of the probability in a Bayesian way it would be more appropriate because information for its value could be incorporated from a physical model (as a prior) and each data point could update our estimation for its new value, as a posterior probability.

3.4 Data Smoothing

As a final step in the pre-processing procedure of a data set with high sampling frequency is useful to smooth the data. This is achieved by implementing a simple moving average (SMA) algorithm. The idea is to smooth the response of our signals in order to capture the important patterns in data while leaving out some noise (Hasselaar,2010) or similarly, in a more common machine learning terminology, increase the signal over noise ratio. For the effective application of the SMA algorithm a proper time window for averaging the data should be selected.

The SMA is an unweighted mean of the previous n data and since we have a constant sampling rate of 1 minute this number n is defining our time window. Three factors should be taken under consideration when choosing the value of n:

i. The dynamics of the vessel. The averaging time window should be short enough to capture environmental loading changes that result in changes to ship performance, but long enough to smoothen natural fluctuations. (Hasselaar,2010)

ii. The application in which the data will be used. In a performance analysis task, we can afford to work with a relatively small but indicative data set because we only need to observe the trends. In a deep learning model, we benefit from a large number of data because they may include more noise but also carry more information which a complex model can extract.

iii. The reduction of uncertainty in exchange for detailed information. If the uncertainty of each measured value can be described by a standard deviation σ, and N readings are taken, the standard deviation over the N data points is reduced by (Coleman and Steele, 1998):

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} \qquad (3.10)$$

Hence, to reduce the standard deviation around the mean by a factor of two, four times more measurements of a (constant remaining) parameter is required.

However, we still do not know what is the exact value of a proper time window but we note that (i) the studied ship is a 200 m long containership which means that it has slow dynamics, in the scale of minutes, (ii) the data will be finally used for the training of a neural network.

We often see averages over 10 or 15-minute time windows in the relevant literature (Senteris, 2018) and (Pedersen & Larsen, 2009). We are not going to reject these values but neither confirm them without investigating the effect of the different averaging time windows on the quality of our produced data set. For this reason, we implement the following methodology.

- Average the data set on different time windows starting from 2 min and up to 20 min.
- Estimate the mean value of the produced data set.
- Estimate the difference (in %) from the original dataset’s mean value.
- Plot this difference for each averaging time window.

The results on three basic parameters of the ship’s propulsion are shown in Figure 3.26. This measure of drift from the original dataset’s mean value is considered correlated with the information loss from the process of averaging.

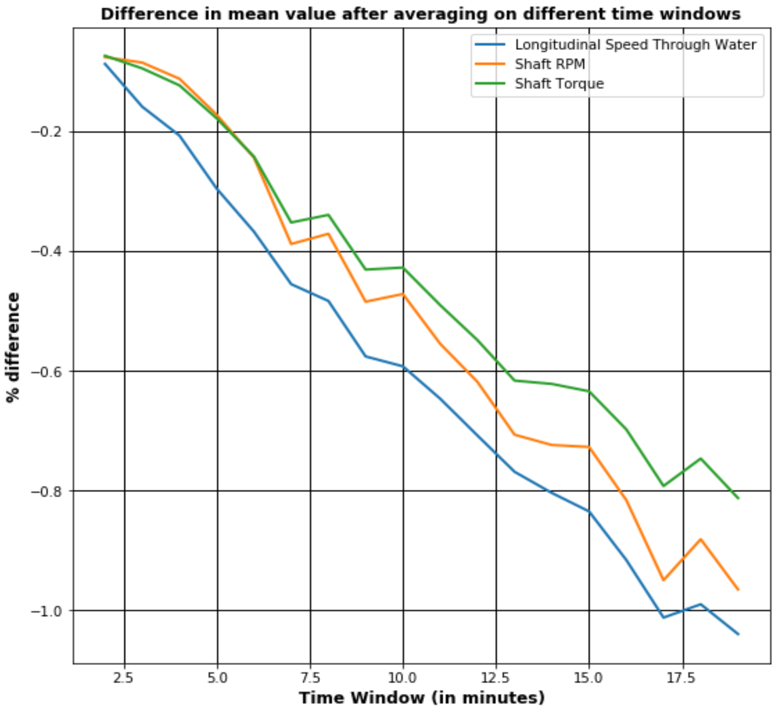


Figure 3.26 The difference between the mean value of the original data and the averaged over different time-windows data. Only three basic parameters of ship's propulsion are plotted, for space economy.

In addition, we calculate the standard deviation in each group of data points that is being averaged and we plot the resulting mean standard deviation (Figure 3.27), for each different averaging time window. The mean standard deviation is considered to be indicative of the information loss because averaging a group of data with zero standard deviation results to zero information loss. For better understanding, an example of this calculation is presented below:

Table 3.3 Twenty consecutively recorded values of the ship's STW (knots).

16.72	16.745	16.775	16.684
16.75	16.736	16.76	16.6
16.738	16.825	16.753	16.592
16.722	16.83	16.76	16.55
16.726	16.712	16.727	16.718

The Table 3.3 has twenty values of the STW from the original data set, which has a 1-minute sampling period. If we average these values on a 4-minute time window we can get five mean values and five standard deviations from the five sets of the four original values. So, we get a pair of a mean value and a standard deviation and if we average the standard deviations from the whole dataset, we get the mean value of the standard deviation. By repeating this process for every different averaging time window, we can produce the diagram of Figure 3.27.

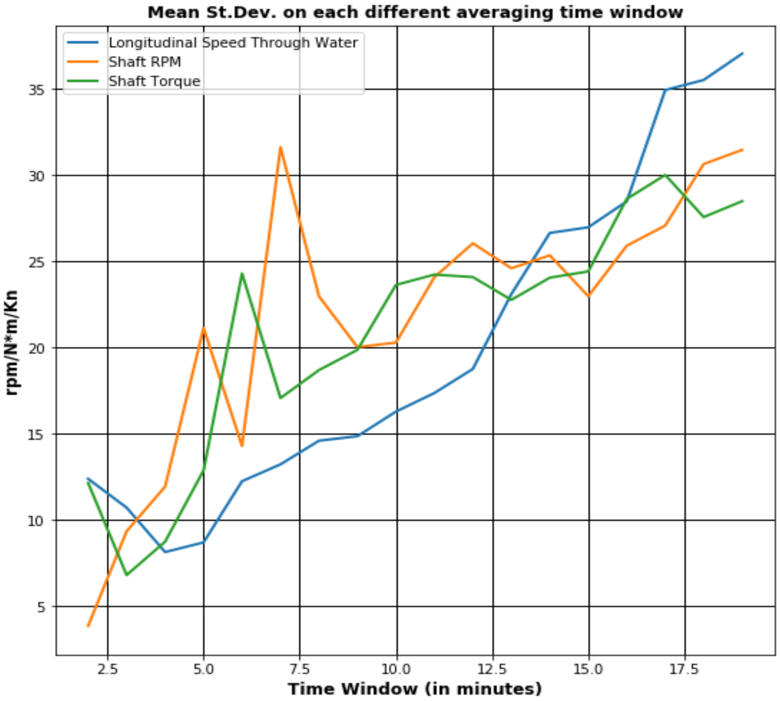


Figure 3.27 The mean standard deviation on each different averaging time window for three basic parameters of ship's propulsion.

After all we confirm what we expected in first place. Increasing the averaging time window results to more information loss and a more distort image of the original data. This is validated in Figure 3.26 as well as in Figure 3.27. In the first because the averaged data set's mean value differs more from the original data set's mean value as the averaging time window increases. In the second because larger values of standard deviation are directly related with larger values of averaging time windows. Finally, we decide to use 5-minute averaging time windows.

An example of the averaged versus the original signal is demonstrated in Figure 3.28. The signal in this figure contains a 5-hour long sample of the ship's STW and the effect of smoothing while keeping enough details is obvious.

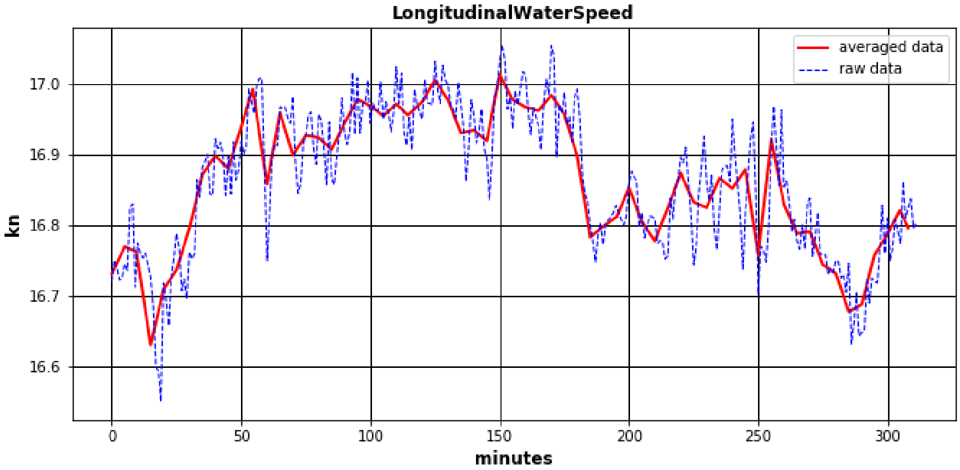


Figure 3.28 A 5-hour sample of the ship's STW plotted on its original form (blue dashed-line) and after being smoothed (red continuous line) with the SMA algorithm on a 5-minute averaging time window.

3.5 Data Quality Control

As it is mentioned in the beginning of this chapter, the entire concept of data pre-processing exists for the evaluation and the improvement of the information contained in the available dataset so as to coincide better with the reality that is faced by the ship. However, it is essential to pay attention to the effect that any filtering, correcting or transforming algorithm acting upon the data, may has on them. It would be pointless if the pre-processing of the data distorts their signals and the information they carry to a degree where they are no longer reliable and realistic. The idea of “careful” data pre-processing is also found in (Aldous,2015) and is summed up in the phrase: “The level of filtering is subjective and requires balance between removing inaccurate data points that will incorrectly skew the results and preserving valuable information about the system physics.”.

The term "data quality" here is used with the notion that differences in statistical parameters (mean value, standard deviation) between the original dataset and the processed one declares poorer data quality. The first attempt to quantify the impact of a pre-processing algorithm on the data was presented in the previous section 3.4. In that case, the only parameter that had to be chosen (the averaging time-window size) for applying the SMA algorithm and smoothen the data, was directly affecting the data quality (see Figure 3.26). In contrast, for the “correcting” and “cleaning” algorithms, that were applied to the data, (section 3.2 and 3.3) no quality metrics have been demonstrated yet, because they involve more parameters. This section is dedicated to investigate the impact of these algorithms on the data.

In regard to the correcting algorithm of Section 3.2, we estimate the percentage of data points that are affected by the algorithm for two different subsets of the dataset. The Figure 3.29 shows the percentages of the affected data points for each parameter, with blue columns for the Subset A that contains 14,000 data points and with orange columns for the Subset B that contains 140,000 data points.

The algorithm was applied in these two subsets, that differ by order of magnitude, in order to investigate if the occurring frequencies of corrections depend on the size of the dataset. From the obtained results, we can assume that the correcting algorithm displays a quite robust behavior when scaling up since when applied to a ten times larger dataset it is affecting the same percentage of data points.

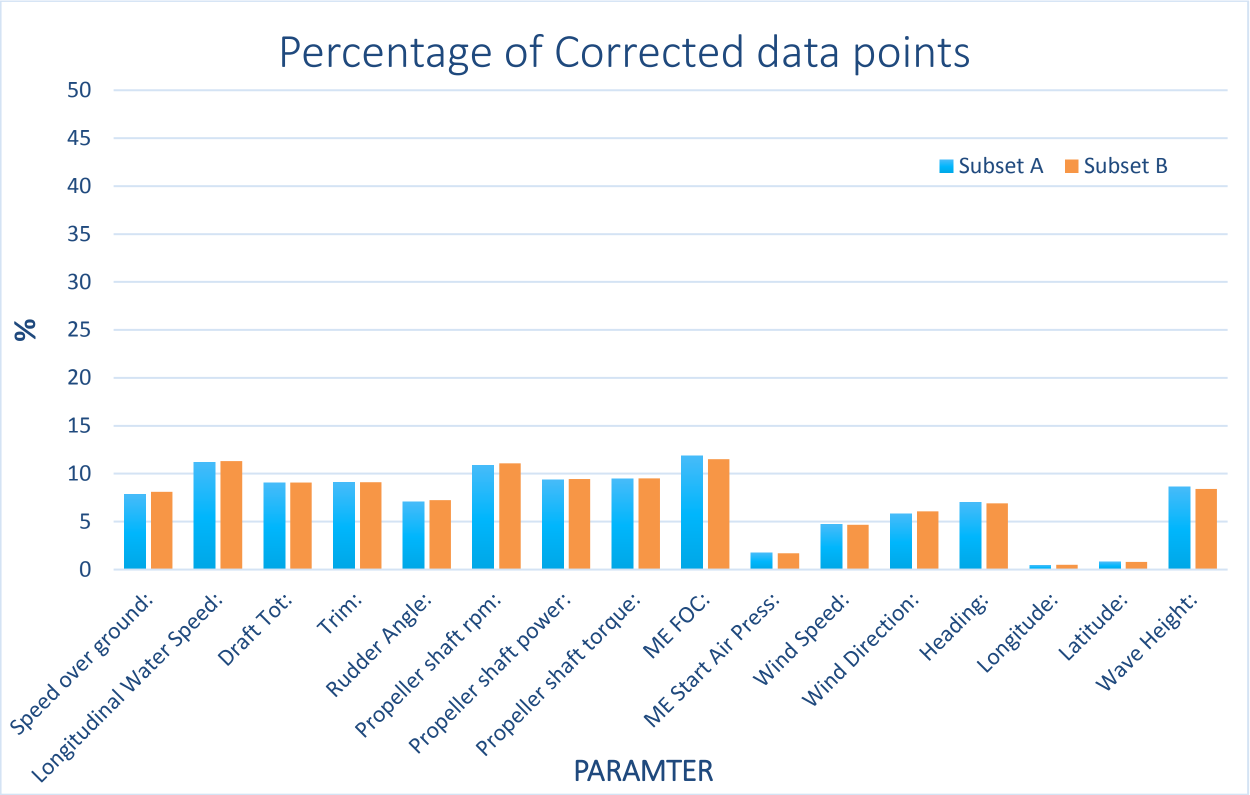


Figure 3.29 The percentages of the affected by the correcting algorithm data points for two subsets of different size.

Encouraging results are also acquired when we calculated the difference in the mean values between the original data set and the “corrected” one. This difference is estimated as a percentage of the original dataset’s mean value. In Figure 3.30 are presented the differences in the mean value of every parameter in the dataset. We notice that the mean values have only so slightly changed that no concerns are raised regarding data distortion phenomena. The largest difference of mean value is observed in the M/E FOC signal but the 0.7396% difference is considered rational since it was noticed to be a signal with extreme values on regular intervals.

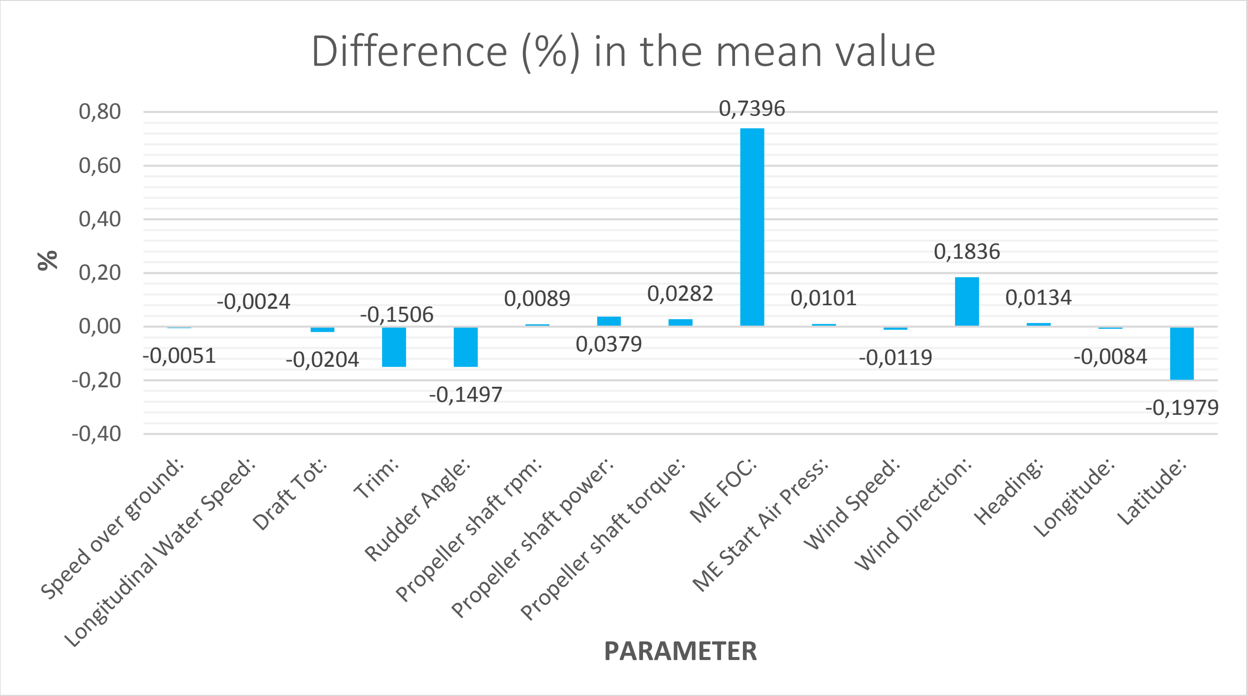


Figure 3.30 The difference in the mean value between the original and the corrected dataset. The differences are estimated as percentage of the original dataset's mean value.

The next algorithm that was applied on the dataset is the "cleaning" algorithm from Section 3.3. Here, a much simpler metric is used to check the rationality of the algorithm. Only the number of the deleted data points is calculated and presented as a percentage of the total number of data points that were examined. In this case, the mean values of the parameters before and after the "cleaning" process are not calculated firstly because a very small portion of the data is affected by the algorithm and secondly because extreme values are almost symmetrically rejected (see Figure 3.24) from the dataset.

Table 3.4 The percentage of data points that were dropped from the dataset as outliers.

Primary parameter →	Propeller Shaft RPM	% of data dropped as outliers
Secondary parameter:		
	Propeller Shaft Torque	0.96 %
	M/E FOC	1.35 %
	Longitudinal Water Speed	1.04 %

At this point, it is stated again that in this case of "cleaning", the entire row of data is dropped from the dataset if an outlier is detected on any parameter of a particular timestamp. In contrast with the previous, "correcting" algorithm, that affects only the data point on the specific parameter that is being tested and does not affect the data on the other parameters. This means that on Table 3.4 the percentages could be summed to estimate the total percentage of data rows removed from the dataset (3.35% in this case) but in Table 5, the percentages cannot be summed for estimating the total percentage of data rows affected.

The data pre-processing procedure concludes with a simple last step which is the filtering for sea current above 1 knot. Data rows where the absolute difference between the SOG and the STW is above 1 knot are deleted. No data quality metric can be introduced for this action because it is actually an operational filter that may additionally reject data points where faulty measurements took place due to GPS or speed log malfunction. Nevertheless, the number of data rows dropped from this filter is 11.23 %. Other cases where the data are filtered for operational criteria (i.e. limits on draft, wind & wave, engine rpm etc.) are not considered to be part of the pre-processing procedure because they are implemented application-wise only.

4 Feature Engineering

Feature engineering is the process of generating, analyzing and finally selecting parameters (= features) to use as inputs in a statistical or machine learning model. Feature engineering is of fundamental interest nowadays since the technology for acquiring large datasets from the Continuous Monitoring of dozens of onboard sensors or metocean data is becoming commonplace. These large datasets, besides from high frequency, are also multiparameter due to the fact that data are transmitted and stored from almost every sensor on the ship. This fact provides the opportunity for the exploitation of a large number of features in the models. In statistical modeling and especially in machine learning when there are many features to choose from, feature engineering is essential.

Therefore, on a modern approach, that aims to provide a state-of-the-art framework for developing data driven ship performance modeling, a chapter dedicated to feature engineering could not be absent. Feature engineering is a relatively abstract term in the machine learning vocabulary. It is considered more a virtue than an analytical skill, as there are no well-established or conventional methods for performing it on an arbitrary machine learning problem. Two indicative descriptions or definitions of the Feature Engineering process are found in Wikipedia (which is used in many other sources) and in Microsoft's Azure Documentation respectively.

- "Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated feature learning. Feature engineering is an informal topic, but it is considered essential in applied machine learning. (Wikipedia, 2019).
- "Feature engineering attempts to increase the predictive power of learning algorithms by creating features from raw data that help facilitate the learning process.
    - **Feature engineering**: This process attempts to create additional relevant features from the existing raw features in the data, and to increase the predictive power of the learning algorithm.
    - **Feature selection**: This process selects the key subset of original data features in an attempt to reduce the dimensionality of the training problem.

  Normally feature engineering is applied first to generate additional features, and then the feature selection step is performed to eliminate irrelevant, redundant, or highly correlated features." (Microsoft, 2019)

In description (a) of the feature engineering it is referred that "The need for manual feature engineering can be obviated by automated feature learning", but automated feature learning or feature extraction is a capability that mainly deep Convolutional Neural Networks (CNNs) have and since CNNs are used in classification problems they are not utilized in the present study. In the case of Feed-Forward Neural Networks (FNNs) that are commonly used in regression problems, the whole procedure is carried out manually.

In description (b) feature engineering is divided into two steps: (i) feature engineering, which is about the creation of additional relevant features from the raw data, and from now on we will call this step *feature generation*, and (ii) feature selection. The implementation of this two-step-approach is demonstrated in this chapter. The goal is mainly to select the features that will produce the best performing model for the ship's propulsion but also to document in general, some of the most popular practices, encountered among data scientists, for feature engineering.

4.1 Feature Generation

Feature generation, as mentioned earlier, is the process of augmenting the number of features of the original dataset. It is a relatively simple procedure but usually requires an insight or intuition about the examined problem, and sometimes imagination could work too. Until this step of the analysis, the discussion about feature engineering has been quite abstract but from now on, it shall focus on the available features and the model that will be built with them.

Firstly, the available features (or parameters) of the dataset are presented in Table 4.1. In section 3.5 the data were smoothed with the use of SMA algorithm, an unweighted moving average of 5-minute time windows. While averaging the data, the standard deviation of these 5-minute time windows is also calculated, as a measure of the steadiness of the ship's condition in each group of data that is averaged. This is the first example of generating features since for every column (feature) in the dataset, a new one is added that holds the values of the standard deviation of the original parameters (see Table 4.2). Similarly, the derivative of some features (where has physical sense) could have been computed but it was obviated due to the co-existence of two deterrent factors: i) the inefficient sampling frequency (not adequately high) in comparison to the ship's heave and pitch motions' dynamics that would lead to rough estimations of the derivatives, and ii) the steadiness of the speed and rpm signals with the presence of noise that would lead to the estimation of the derivative of the noise rather than the derivative of the actual measured physical quantity.

Table 4.1 The labels of the original features of the dataset.

Speed Over Ground	Longitudinal Water Speed	Draft Tot	Trim	Rudder Angle
Propeller shaft rpm	Propeller shaft power	Propeller shaft torque	ME FOC	M/E START AIR PRESS
Wind Speed	Wind Direction	Heading	Longitude	Latitude

Table 4.2 The labels of the newly generated features. The "STD" prior to the names of the features stands for the term "Standard Deviation".

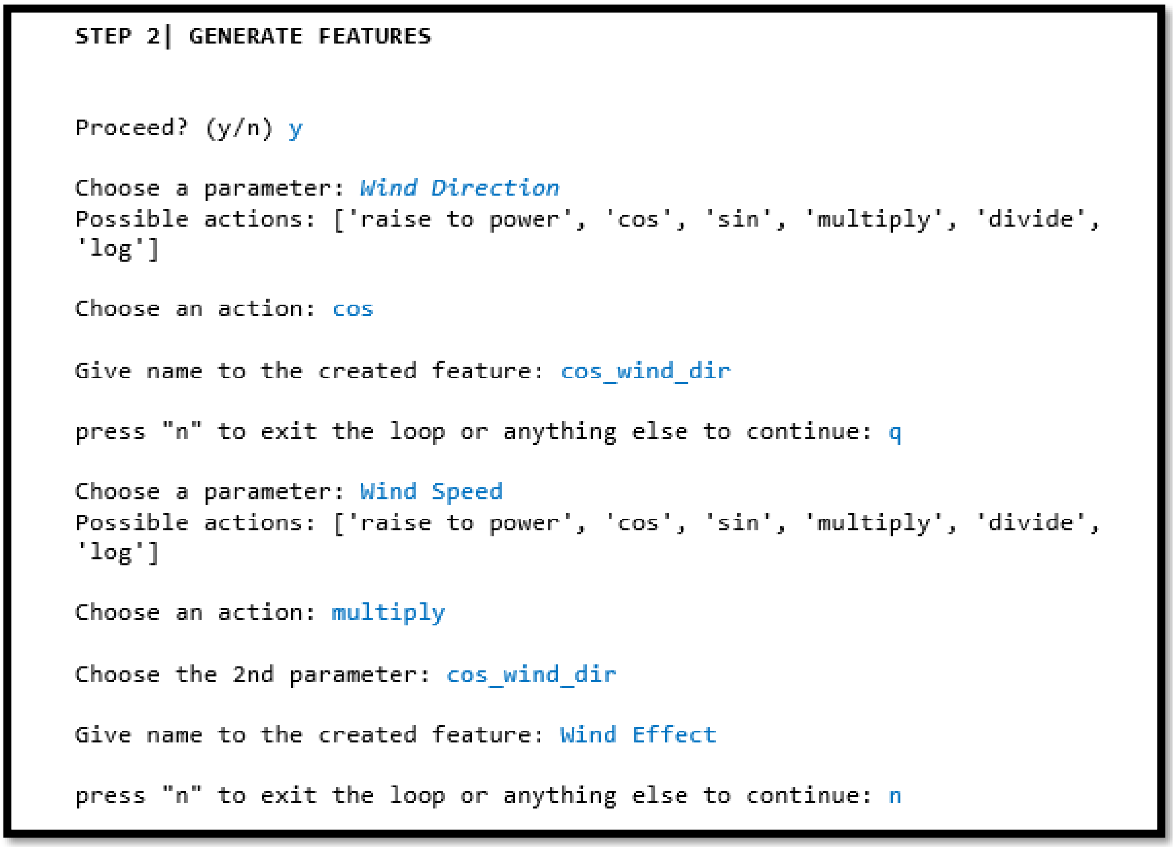
STD_Speed Over Ground	STD_Longitudinal Water Speed	STD_Draft Tot	STD_Trim	STD_Rudder Angle
STD_Propeller shaft rpm	STD_Propeller shaft power	STD_Propeller shaft torque	STD_ME FOC	STD_M/E START AIR PRESS
STD_Wind Speed	STD_Wind Direction	STD_Heading	STD_Longitude	STD_Latitude

Other ways that new features can be produced have to do with applying certain mathematical operations, functions or transformations on the existing features. For example:

i) Add or subtract two columns.
ii) Multiply or divide two columns.
iii) Raise to the power of α.
iv) Apply a trigonometric function (sin, cos, tan etc.).
v) Apply a logarithmic function (log, ln etc.)
vi) Any combination of the above.

Even the absolute value of an existing feature can be regarded as a new feature.

In order to achieve a systematic approach, simplify and speed up the feature generation procedure a computer program was developed that generates features interactively. By receiving commands from the keyboard, it generates the desired feature, names it and stores it in the dataset next to the rest features. The dialogue on the computer monitor is presented below in Figure 4.1:



```
STEP 2| GENERATE FEATURES


Proceed? (y/n) y

Choose a parameter: Wind Direction
Possible actions: ['raise to power', 'cos', 'sin', 'multiply', 'divide',
'log']

Choose an action: cos

Give name to the created feature: cos_wind_dir

press "n" to exit the loop or anything else to continue: q

Choose a parameter: Wind Speed
Possible actions: ['raise to power', 'cos', 'sin', 'multiply', 'divide',
'log']

Choose an action: multiply

Choose the 2nd parameter: cos_wind_dir

Give name to the created feature: Wind Effect

press "n" to exit the loop or anything else to continue: n
```

Figure 4.1 Feature generation dialogue from the interactive feature generation program that generates and stores the desired new feature. The example of generating the "Wind Effect" parameter. With blue color are the user's inputs.

In the example of Figure 4.1, the data points of the Wind Direction parameter are transformed from degrees in the range [0,360] to real numbers in the range [-1,1] with the use of the trigonometric function cosine. The transformed parameter is stored as a new feature under the name *cos_wind_dir* and is then multiplied (element-wise) by the data points of the Wind Speed parameter in order to give another new feature named *Wind Effect*. The physical meaning of the new feature is that the effect of the wind in the ship's resistance is maximized when it is a headwind, minimized when it is a tailwind and symmetrical when it is a lateral wind. Of course, the added resistance due to wind is not zero when the wind blows perpendicular to the hull and in the context of Artificial Neural Networks (ANNs) a zero input does not imply a zero effect on the output variable, but this is not a discussion for this section.

Another feature that was generated is the “sea current” which is simply the result of the subtraction of the STW from the SOG. Here it should be mentioned that a feature could be generated just for providing a better insight on the problem or to assist in filtering the data and not necessarily to be included in the final model. That was the case for the “sea current” feature.

Last but not least, the most delicate case of feature generation was the one required for generating a feature that is labeled as “Fouling”. It should be clear that this is not an effort to describe qualitatively or to quantify the actual hull fouling. The goal is to provide some temporal information to the model because it is obvious that during the 12-month period that the training data were gathered the ship’s hull and propeller condition could not have remain stable (normally noticeable fouling appears in a year’s period). By introducing a feature that is time-depended the model will be able to discriminate the data points that were collected earlier on time from those that were collected later on. The feature is generated by the following equation:

$$F_i = \log\left(7000 + \sqrt{18 \cdot t_i}\right) - 3 \qquad (4.1)$$

Where,

- t_i is the elapsed time with respect to the first data point.

The constants and the multiplier in equation (4.1) are there to adjust the rate of change and the initial and final value of the F_i. Because the independent variable t_i of the function F_i (see equation (4.1) has a strictly defined domain due to the way that is being generated (from the dataset’s timestamps), the only way to manipulate the outputs of the function is by the introducing these constants and a multiplier. Their final values are chosen after iterative testing and are those that minimize the model’s error.

Finally, the total number of features that was generated from the initially 16, was 37 and they are presented in Table 4.3. However, these 37 features should be investigated for their properness to be included in a machine learning model since some may be co-linear with others or redundant and that is the object of the next section.

Table 4.3 The labels of the initial and the generated features.

Speed Over Ground	Longitudinal Water Speed	Draft Tot	Trim	Rudder Angle
Propeller shaft rpm	Propeller shaft power	Propeller shaft torque	ME FOC	M/E START AIR PRESS
Wind Speed	Wind Direction	Heading	Longitude	Latitude
Wave Height	Sea Current	Fouling	Cos_wind_dir	Wind Effect
STD_Speed Over Ground	STD_Longitudinal Water Speed	STD_Draft Tot	STD_Trim	STD_Rudder Angle
STD_Propeller shaft rpm	STD_Propeller shaft power	STD_Propeller shaft torque	STD_ME FOC	STD_M/E START AIR PRESS
STD_Wind Speed	STD_Wind Direction	STD_Heading	STD_Longitude	STD_Latitude
STD_Wave Height	STD_Sea Current			

4.2 Feature Selection

In general, feature selection is not a necessary step for every machine learning application. When a limited number of features is available for a problem, discarding some of them will not improve either the model's performance nor the computational cost significantly. However, it is quite frequent to have an abundance of data and features for the single target that we aim to predict. In a favorable scenario, reducing the number of features could either reduce the computational cost while affecting marginally the performance of the model or improve both. For example, when dropping redundant or confusing (with excessive noise in the signal) features from the model, its accuracy will rise while fewer computations will be needed.

Particularly, in this section, the features that are presented in section 4.1 will be evaluated with respect to their potential contribution to the performance of an FNN (see section 5.1 for a description of FNN) regression model. Frequently the term “predictor” or “estimator” is used when we refer to a feature that is used in machine learning model as an input and contributes in predicting or estimating the output’s value correctly. Therefore, the aim of this section is to investigate which features are “good predictors” for a particular target. Target or targets are the variable(s) that are selected as outputs of the model.

Firstly, it should be clear how the number of features in a model relates to the complexity or the degree of the model. In a FNN the number of inputs is always equal to the number of features that are exploited by the model. Each input is accompanied by a number of trainable parameters that should be learned by the model in the training process. The total number of trainable parameters in the model corresponds to the degree or the complexity of the model, in the sense that each extra trainable parameter in the model offers an extra degree of freedom to the derived hyper-surface (because we deal with much more than three dimensions) that could interpolate a complex dataset more efficiently. For example, a dataset with strongly non-linear relations, among its parameters and the target, would require a complex model for successful interpolation.

The evaluation of the performance of a feature as a predictor in a machine learning model could be based on theoretical-physical knowledge or statistical evidence among the examined feature and each target variable. This is essential, in order to systematically select features without testing the actual model repetitively and save time. However, it is always possible to train and test a model with and without the existence of a feature in the inputs in order to directly decide if it is a good predictor.

Prior to the statistical analysis and if there is not any explicit theoretical law involving the candidate feature and the target variable, the below questions need be asked:

(a) Does the feature provide any valuable information or it seems redundant with respect to the target variable?
(b) Is it highly correlated to any other feature and provides the exact same information to the model?
(c) Would the increase in the solution-input space dimension benefit the model or it already seems too complex for the problem that is being addressed?

While considering the above it is important to keep in mind that thanks to the abstract inferring capabilities of the ANNs a feature does not have to be accurate in a physical and quantifying sense, in order to be a good predictor and improve the model’s performance. However, when the general and qualitative discussion is completed, metrics and statistics about the data should be calculated and evaluate the features based on these results as well.

In the following paragraphs the feature selection procedure that was followed for creating an FNN model that estimates the main engine's Fuel Oil Consumption (ME FOC) and another one that estimates the Ship's Speed Through Water (STW) is demonstrated.

Question (a) cannot be addressed with the use of statistics but we make a first selection of features based on our physical understanding on the problem. For the question (b) though, is trivial to find an answer. The correlation among all the parameters has been investigated in general, in Section 3.1, where parametric plots were also produced in order to visually inspect the data for non-linear types of correlation. Here the linear correlation coefficient (Pearson product-moment) for the selected features of each model is estimated, only with respect to the target variable (ME FOC or Longitudinal Water Speed). The results are shown in Table 4.4. Because the selected features should not be highly correlated with the target variable, a feature rejection threshold could be set for the value of the correlation coefficient. In contrast, it is not worrying if the coefficient value is almost zero since it only declares that there is no linear correlation.

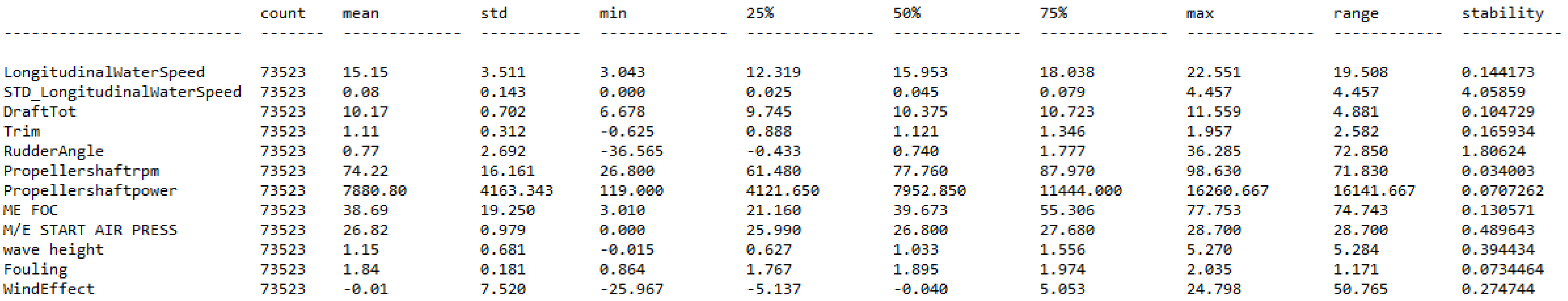
From the obtained values in Table 4.4, there are only a few that give us reason to worry about the "properness" of some feature. These are the very high values of the linear correlation coefficient for the "Propeller shaft power" (for both targets) and the "Propeller shaft rpm". These values are by themselves a strong argument for the rejection of the referred features but the nature of the problem is not allowing us to take decisions based on statistics only. These two features do not get rejected yet, because it would be impractical to have a model that can predict the speed or the consumption of a ship without having the ability to provide the engine's power or the propeller's rpm as input.

Table 4.4 Table of linear correlation coefficient values between input features and target variables.

CORRELATION TABLE (input features linear correlation with the target variable)		
	Longitudinal Water Speed	**ME FOC**
Longitudinal Water Speed		0.939
STD_Longitudinal Water Speed	-0.233	-0.163
Draft Tot	0.189	0.213
Trim	0.493	0.450
Rudder Angle	-0.147	-0.153
Propeller shaft power	0.935	0.974
Propeller shaft rpm	0.998	0.999
M/E START AIR PRESS	-0.00316	-0.0110
Wave Height	0.0463	0.180
Fouling	0.0265	0.0650
Wind Effect	-0.00240	0.00310

The evaluation of the features is supplemented with the calculation of another set of statistics that describes them. The results are presented in Table 4.5.

Table 4.5 Description of the data in each feature via the calculation of a set of statistics.



```
                           count    mean          std         min           25%           50%           75%           max           range        stability
-------------------------  -------  ------------  ----------  ------------  ------------  ------------  ------------  ------------  -----------  ----------

LongitudinalWaterSpeed     73523    15.15         3.511       3.043         12.319        15.953        18.038        22.551        19.508       0.144173
STD_LongitudinalWaterSpeed 73523    0.08          0.143       0.000         0.025         0.045         0.079         4.457         4.457        4.05859
DraftTot                   73523    10.17         0.702       6.678         9.745         10.375        10.723        11.559        4.881        0.104729
Trim                       73523    1.11          0.312       -0.625        0.888         1.121         1.346         1.957         2.582        0.165934
RudderAngle                73523    0.77          2.692       -36.565       -0.433        0.740         1.777         36.285        72.850       1.80624
Propellershaftrpm          73523    74.22         16.161      26.800        61.480        77.760        87.970        98.630        71.830       0.034003
Propellershaftpower        73523    7880.80       4163.343    119.000       4121.650      7952.850      11444.000     16260.667     16141.667    0.0707262
ME FOC                     73523    38.69         19.250      3.010         21.160        39.673        55.306        77.753        74.743       0.130571
M/E START AIR PRESS        73523    26.82         0.979       0.000         25.990        26.800        27.680        28.700        28.700       0.489643
wave height                73523    1.15          0.681       -0.015        0.627         1.033         1.556         5.270         5.284        0.394434
Fouling                    73523    1.84          0.181       0.864         1.767         1.895         1.974         2.035         1.171        0.0734464
WindEffect                 73523    -0.01         7.520       -25.967       -5.137        -0.040        5.053         24.798        50.765       0.274744
```

In Table 4.5 the first column refers to the number of data points in each feature, the second to the mean value, the third to the standard deviation, the fourth and the eighth to the minimum and maximum contained value, respectively. The three in-between columns are the 1st, 2nd (median value) and 3rd quartiles[2], respectively and the ninth column is the range of each feature (maximum-minimum). The last column, stability, is a measure of data points' concertation around the mean value of the feature or in other words, how static a feature is. It is modelled as the percentage of data points that lie only 0.1% of the standard deviation away from the mean value. The stability column could be useful in case of a feature, like a ship's draft, being almost constant (i.e. design draft for all the available data) and so it needs to be rejected because it will not provide any information to the model. In our case, we observe that only the standard deviation of the STW has a relatively high stability value but this is accepted because operationally the ships speed stays quite steady for long periods. The same is true for the rudder angle, which perturbates around 0 degrees of angular displacement when the ship sails steadily.

[2] A percentile is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations fall. For example, the 25th percentile is the value below which 25% of the observations may be found. The 25th percentile is also known as the first quartile (Q1), the 50th percentile as the median or second quartile (Q2), and the 75th percentile as the third quartile (Q3).

Finally, at this point, we should either decide to move forward with the existing features or proceed to actually test the features based on the performance of the FNN model. In this way question (c) could be answered as well.

In summary, the process of feature engineering is: (by Prof. Ryan Baker in Coursera)

I. Brainstorming or Testing features;
II. Deciding what features to create;
III. Creating features;
IV. Checking how the features work with your model;
V. Improving your features if needed;
VI. Go back to brainstorming/creating more features until the desired model performance is achieved.

However, implementing an FNN model and testing the different possible sets of features requires the fine-tuning of other, irrelevant to the features, parameters because is pointless to test the features on an inappropriate model. For this reason, quantitative results regarding the performance of the model when some features are included or excluded from the model will be presented in the next chapter.

4.3 Principal Component Analysis

So far efforts have been applied in the direction of generating and selecting features based on the physical understanding of the problem and the statistical characteristics of the features. There is, however, a more analytical and abstract way for accomplishing the afore mentioned procedures. The well-known method of Principal Components Analysis (PCA) is often used in the feature extraction procedure. But the results of this kind of feature extraction method are not guaranteed and of course do not apply to any type of dataset or problem.

Principal Components Analysis is an unsupervised linear transformation technique that is widely used across different fields and has been successful in tasks like extracting features and reducing dimensionality in high-dimensional data, like images. Other popular applications of PCA include exploratory data analysis, de-noising of signals in stock market trading, and the analysis of genome data and gene expression levels in the field of bioinformatics. (Raschka, 2015)

As we find in (Bishop, 2006)there are two commonly used definitions of PCA that give rise to the same algorithm. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized (Hotelling, 1933). Equivalently, it can be defined as the linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and the projections (Pearson, 1901). Figure 4.2 explains these two definitions with a simple example.

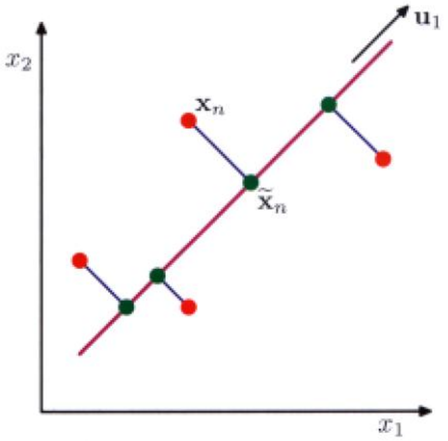


Figure 4.2 Principal Components analysis seeks a space of lower dimensionality, known as the principal subspace and denoted by the magenta line, such that the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors, indicated by the blue lines. (Source: Bishop,2008)

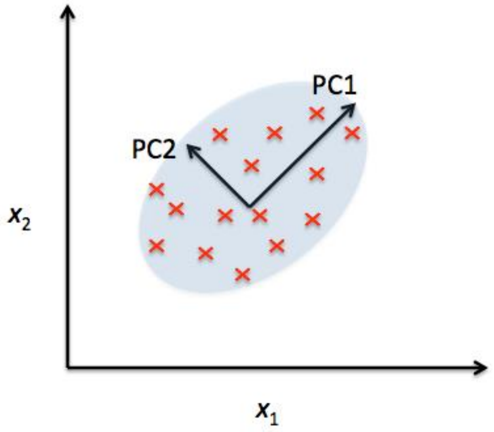


Figure 4.3 The axes x_1 and x_2 are the original feature axes, and PC1 and PC2 are the maximum variance axes or the principal components. (Source: Raschka,2015)

Another explanatory graph and a practical description of the PCA is found in (Raschka, 2015). The aim of the PCA is to find the directions of maximum variance in high-dimensional data and project them onto a new subspace of equal or lower dimension than the original. The orthogonal axes (principal components) of the new subspace can be interpreted as the directions of maximum variance given the constraint that the new feature axes are orthogonal to each other as illustrated in Figure 4.3.

However, the question that remains is how to convert from the original axes to the principal components in order to apply dimensionality reduction by truncating the input vector. To achieve this, we shall find an invertible linear transformation T such that the truncation of the input vector is optimum in the mean-square-error sense. The following analysis is summary of the method based on (Haykin, 2009).

i. Mathematical formulation

Let X denote an *m*-dimensional random vector representing the features of a model. We normalize the X vector so that,

$$\mathrm{E}[\mathbf{X}] = 0$$

Let q denote a unit vector of dimension *m*, onto which the vector X is to be projected and A is the projection.

$$\boldsymbol{A} = \boldsymbol{X}^T\boldsymbol{q} = \boldsymbol{q}^T\boldsymbol{X} \qquad (4.1)$$

The projection A is a *random variable* and since the random vector X has zero mean, it follows that:

$$\mathrm{E}[\mathbf{A}] = \boldsymbol{q}^T\mathrm{E}[\mathbf{X}] = 0$$

In addition, the variance of A is the same as its mean-square value, so we write

$$\sigma^2 = E[\boldsymbol{A}^2] \overset{(4.1)}{\Longrightarrow}$$

$$E[\boldsymbol{A}^2] = E[(\boldsymbol{X}^T\boldsymbol{q})(\,\boldsymbol{q}^T\boldsymbol{X})]$$

$$= \boldsymbol{q}^T E[\boldsymbol{X}^T \boldsymbol{X}]\boldsymbol{q} = \boldsymbol{q}^T \boldsymbol{R}\boldsymbol{q} \qquad (4.2)$$

The *m-by-m* matrix **R** is by definition the *correlation matrix* of the random vector **X**.

We observe that the correlation matrix **R** is symmetric, which means that

$$\boldsymbol{R} = \boldsymbol{R}^T$$

From this property, it follows that if **α** and **b** are any *m-by-1* vectors, then

$$\boldsymbol{a}^T \boldsymbol{R}\boldsymbol{b} = \boldsymbol{b}^T \boldsymbol{R}\boldsymbol{a} \qquad (4.3)$$

From Equation (4.2), we see that the variance $\boldsymbol{\sigma}^2$ of the projection **A** is a function of the unit vector **q**; we may thus write

$$\psi(\boldsymbol{q}) = \sigma^2 \overset{(4.2)}{\Longrightarrow}$$

$$\sigma^2 = \boldsymbol{q}^T \boldsymbol{R}\boldsymbol{q} \qquad (4.4)$$

And hence consider $\boldsymbol{\psi(q)}$ as a *variance probe*.

The next issue to be considered is that of finding those unit vectors **q** along which $\boldsymbol{\psi(q)}$ has *extremal or stationary values* (i.e., local maxima or minima), subject to a constraint on the Euclidean norm of **q**. The solution to this problem lies in the *eigenstructure* of the correlation matrix **R**. If **q** is a unit vector such that the variance probe $\boldsymbol{\psi(q)}$ has an extremal value, then for any small perturbation δ**q** of the unit vector **q**, we find that, to a first order in δ**q**,

$$\psi(\boldsymbol{q} + \delta\mathbf{q}) = \psi(\boldsymbol{q}) \qquad (4.5)$$

Now by substituting $\boldsymbol{\psi(q)}$from Equation (4.4) in Equation (4.5) we get

$$\psi(\boldsymbol{q} + \delta\mathbf{q}) = (\boldsymbol{q} + \boldsymbol{\delta q})^T \boldsymbol{R}(\boldsymbol{q} + \boldsymbol{\delta q})$$

After executing the multiplications on the right-hand side and ignoring the second-order terms, we have

$$\psi(\boldsymbol{q} + \delta\mathbf{q}) = \boldsymbol{q}^T \boldsymbol{R}\boldsymbol{q} + \mathbf{2}(\boldsymbol{\delta q})^T \boldsymbol{R}\boldsymbol{q}$$

$$= \psi(\boldsymbol{q}) + \mathbf{2}(\boldsymbol{\delta q})^T \boldsymbol{R}\boldsymbol{q}$$

$$=> \psi(\boldsymbol{q}) = \psi(\boldsymbol{q}) + \mathbf{2}(\boldsymbol{\delta q})^T \boldsymbol{R}\boldsymbol{q}$$

$$= (\boldsymbol{\delta q})^T \boldsymbol{R}\boldsymbol{q} = \mathbf{0} \qquad (4.6)$$

Just any perturbations δ**q** of **q** are not admissible; rather, we are restricted to use only those perturbations for which the Euclidean norm of the perturbed vector (**q** + δ**q**) remains equal to unity; that is,

$$(\mathbf{q} + \delta\mathbf{q})^T (\mathbf{q} + \delta\mathbf{q}) = 1$$

Hence, to a first order approximation in δ**q**, we require that

$$(\boldsymbol{\delta q})^T \boldsymbol{q} = \mathbf{0} \qquad (4.7)$$

This means that the perturbations δ**q** must be orthogonal to **q**, and therefore only a change in the direction of **q** is permitted.

Combining Equations (4.6) and (4.7) requires to scale the dimensionless unity vector **q** with a scaling factor λ with the same dimensions as the entries in the correlation matrix **R**. We may then write

$$(\delta \boldsymbol{q})^T \boldsymbol{R} \boldsymbol{q} = \lambda (\delta \boldsymbol{q})^T \boldsymbol{q} =>$$

$$(\delta \boldsymbol{q})^T (\boldsymbol{R} \boldsymbol{q} - \lambda \boldsymbol{q}) = 0 \qquad (4.8)$$

Equation (4.8) yields that

$$\boldsymbol{R} \boldsymbol{q} = \lambda \boldsymbol{q} \qquad (4.9)$$

This is the equation that governs the unit vectors q for which the variance probe $\psi(\boldsymbol{q})$ has extremal values. It is clear now that Equation (4.9) is a typical *eigenvalue problem* and since the **R** matrix is symmetrical, its eigenvalues are real and nonnegative values. Let the eigenvalues of the *m-by-m* matrix R be denoted by $\lambda_1, \lambda_2, \dots, \lambda_m$ and the associated eigenvectors be denoted by $q_1, q_2, \dots, q_m$ respectively. In a compact form we write

$$\boldsymbol{R} \boldsymbol{q}_j = \lambda_j \boldsymbol{q}_j, \qquad j = 1,2, \dots, m \qquad (4.10)$$

Let the corresponding eigenvalues be arranged in decreasing order as

$$\lambda_1 > \lambda_2 > \cdots > \lambda_j > \cdots > \lambda_m$$

so that $\lambda_1 = \lambda_{max}$. Let the associated eigenvectors be used to construct the *m-by-m* matrix,

$$\boldsymbol{Q} = [q_1, q_2, \dots, q_j, \dots, q_m]$$

We may then combine the set of *m* equations represented in Equation (4.10) into the single equation

$$\boldsymbol{R} \boldsymbol{Q} = \boldsymbol{Q} \boldsymbol{\Lambda} \qquad (4.11)$$

Where $\boldsymbol{\Lambda}$ is a diagonal matrix defined by the eigenvalues of matrix **R**, and the matrix **Q** is an *orthogonal* (orthonormal and unitary) matrix. The fact that matrix **Q** is orthogonal requires distinct eigenvalues.

At this point, we have everything it takes to express the matrix **R** in terms of its eigenvalues and eigenvectors as

$$\boldsymbol{R} = \sum_{i=1}^{m} \boldsymbol{\lambda}_i \, \boldsymbol{q}_i \boldsymbol{q}_i^T = \mathbf{Q} \mathbf{\Lambda} \boldsymbol{Q}^T \qquad (4.12)$$

which is referred to as the *spectral theorem*. Equation (4.12) represents the *eigen-decomposition* of the matrix **R** and is basically equivalent to Principal Components Analysis.

In summary, the eigen vectors of the correlation matrix **R** pertaining to the zero-mean random vector X define the unit vectors $\boldsymbol{q}_j$, representing the principal directions along which the variance probes $\psi(\boldsymbol{q}_j)$ have their extremal values. The associated eigenvalues define the extremal values of the variance probes $\psi(\boldsymbol{q}_j)$.

Finally, we shall explain how the original data points are represented in terms of the principal components. Let the data vector **x** denote a realization (i.e., sample value) of the random vector **X**. Let **α** denote a realization of the random variable **A**. With *m* possible solutions for the unit vector **q**, we find that there are *m* possible projections of the data vector **x** to be considered. Specifically, from Equation (4.1), we note that

$$\boldsymbol{a}_j = \boldsymbol{x}^T \boldsymbol{q}_j = \boldsymbol{q}_j^T \boldsymbol{x}, \quad j = 1,2, \dots . m \qquad (4.13)$$

where the $\boldsymbol{a}_j$ are the projections of **x** onto the principal directions represented by the unit vectors $\boldsymbol{q}_j$.The $\boldsymbol{a}_j$ are called the principal components; they have the same physical dimensions as the data vector **x**. The formula in Equation (4.13) may be viewed as one of *analysis*.

To reconstruct the original data vector x exactly from the projections $\boldsymbol{a}_j$, we proceed as follows: First, we combine the set of projections into a single vector, as shown by

$$\begin{aligned} \boldsymbol{a} &= [a_1, a_2, \dots, a_j, \dots, a_m]^T \\ &= [\boldsymbol{x}^T \boldsymbol{q}_1, \boldsymbol{x}^T \boldsymbol{q}_2, \dots, \boldsymbol{x}^T \boldsymbol{q}_m] \\ &= \boldsymbol{Q}^T \boldsymbol{x} \qquad (4.14) \end{aligned}$$

Next, we multiply both sides of Equation (4.14) by the matrix Q from the left-hand side and due to the fact that matrix **Q** is orthogonal we get the Equation (4.15) for reconstructing the original data vector **x.**

$$\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{a} = \sum_{j=1}^{\boldsymbol{m}} \boldsymbol{a}_j \boldsymbol{q}_j \qquad (4.15)$$

The unit vectors $\boldsymbol{q}_j$ represent a basis of the data space and Equation (4.15) may be viewed as the formula for *synthesis*.

Now that the invertible linear transformation T, that we were looking for is found, the technic of analysis and synthesis can be applied to project the data onto their principal components and perform dimensionality reduction in an optimum, in terms of mean-square-error, sense. Since the matrix **Q** that transforms the original data coordinates is constructed with a decreasing order of eigenvalues magnitude, discarding its last rows leads to an optimal reduction of dimension or de-noising of a signal.

ii. Application in real data

An application of PCA on the obtained ship dataset demonstrates the way that the analysis and synthesis process behave when the degree of dimensionality reduction is a free parameter. The implemented procedure is the following:

- Retrieve a slice of the dataset that has been through the manual feature engineering procedure. In this case it includes 10,000 data vectors (or data points, is used with the same meaning here) and 16 features.
- Normalize the dataset values by turning them to z-score values (zero mean and unit variance).
- Perform the PCA as explained in 4.3.1, in order to obtained the matrix **Q**. (analysis)
- Re-compose the data points by projecting them on the new base (matrix **Q**), the principal components' one. (synthesis)
- Repeat the process of the re-composition, and on each iteration reduce the dimension of the new base by discarding the last eigenvectors (one at a time).

The results of this parametric dimensionality reduction process are presented in Figure 4.4 for a specific set of parameters. The propeller shaft power is plotted against the propeller shaft rpm and the original data points are compared with the transformed ones, in the lower dimension space. The number of components that are used in the re-composition of the data points increases by one as we move from up and left to right and down. The inability to grasp the non-linear behavior of the data is quite clear in the lower dimension cases but after the 11 components, the projection of the transformed data describes accurately the original data variation.

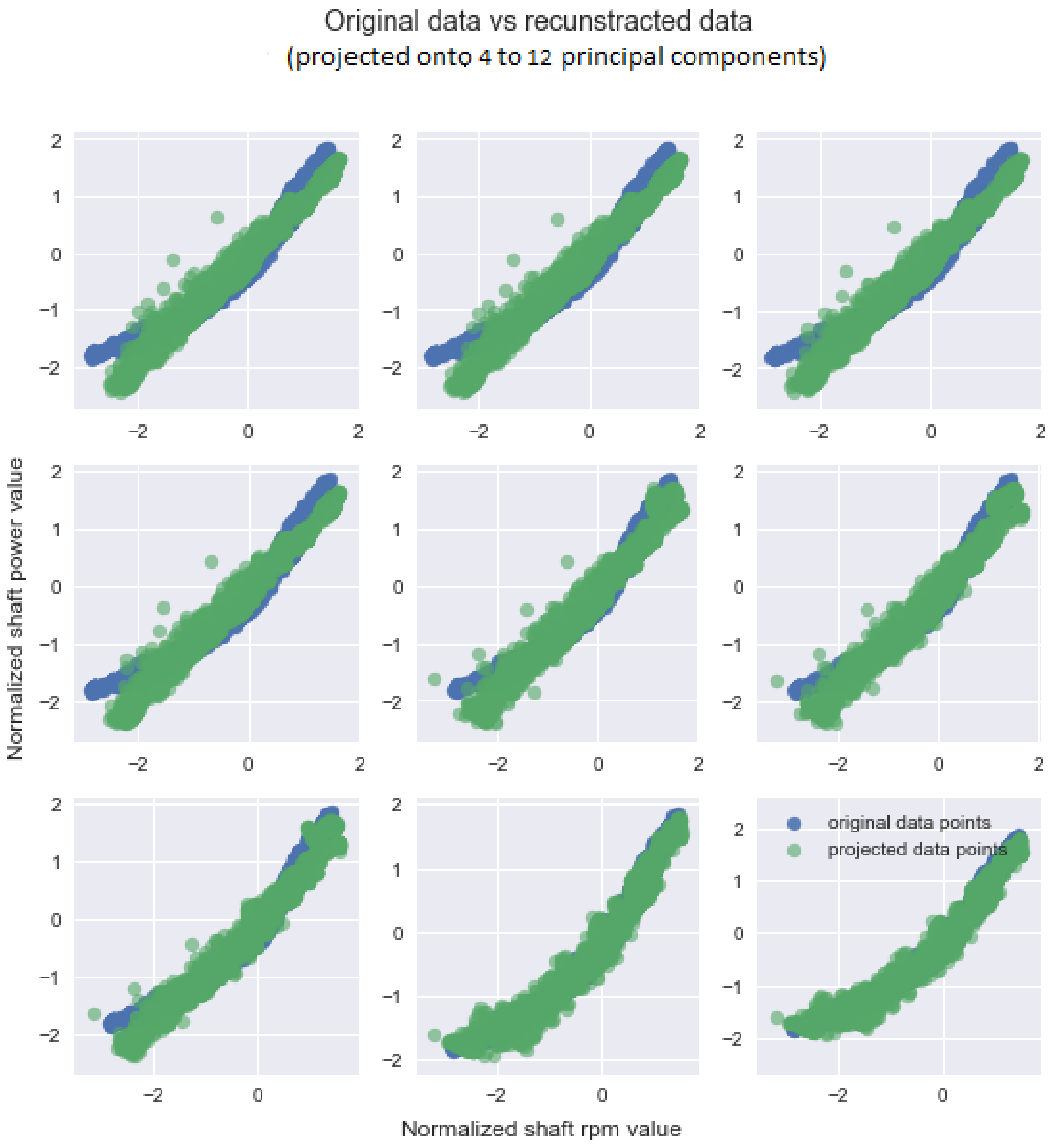


Figure 4.4 The original data points of the propeller shaft power and propeller shaft rpm are plotted against the projected on a lower dimension space, data points. The number of principal components used for the projection are 5 on the top left corner and increase by one as we move left and down.

The above observation is supported by the cumulative explained variance graph in Figure 4.5. Here, the amount of the total variance that is explained by each number of principal components is plotted. The green line demonstrates the 99% threshold, which means that with 10 principal components we can explain the 99% of the original data set variance. That is why in Figure 4.4 the 11 components projection is almost identical to the original data set. Note that this is just one case of parameters that are described satisfactory with fewer dimensions. In other pairs of parameters there is the probability to encounter slightly different behaviors.

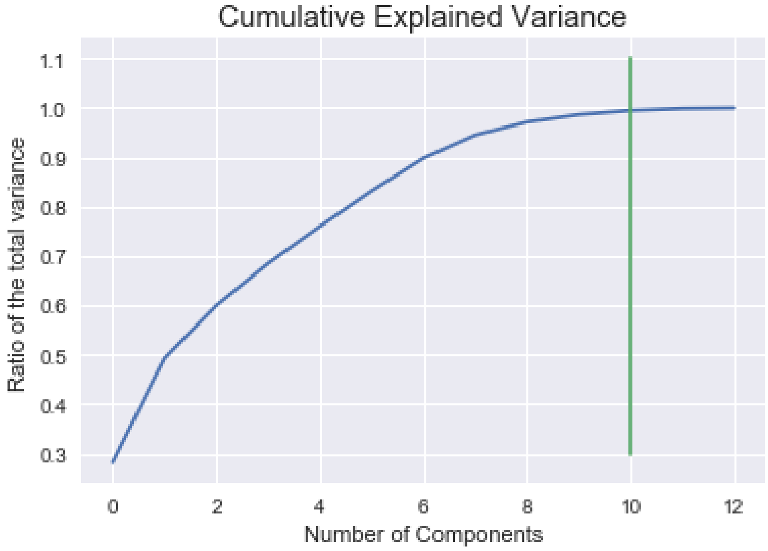


Figure 4.5 The cumulative explained variance by the principal components is a curve that provides useful information regarding the possibility of reducing the dimension of the dataset without loss of valuable information.

The exact same procedure was applied to larger slices of the dataset in order to investigate the computational cost of the method. The findings are presented in Table 4.6. The linear scaling of the computational cost implies that this method is appropriate for use in large datasets.

Table 4.6 The computational time required for applying PCA on datasets of different size

Number of data points	Computational time (seconds)
10,000	1.135
20,000	2.259
40,000	4.598
70,000	8.013
Linear correlation coefficient value	$R^2 = 0.999$

On a last example, principal components analysis may also be applied to a single pair of parameters. The axes of maximum variation among these parameters are calculated in this process and some additional insight for their correlation might be provided. For instance, in Figure 4.6 two pairs of parameters are plotted on their original axes and their principal components are illustrated as well. We notice how the eigenvectors have similar magnitude (which is defined by the value of the eigenvectors) and are almost perpendicular when the data points are scattered all over the original axes plane. In contrast, when the data points in the scatter plot behave in a more ordered way (i.e. when a linear, quadratic, etc. relationship exist among them) the eigenvalues differ by order of magnitude and the eigenvectors form a much smaller angle. Therefore, this example validates the fact that vast differences in the magnitude of the eigenvalues imply that the data variation could be explained satisfactorily even if the last, of the decreasingly sorted eigenvectors, is discarded (dimensionality reduction).

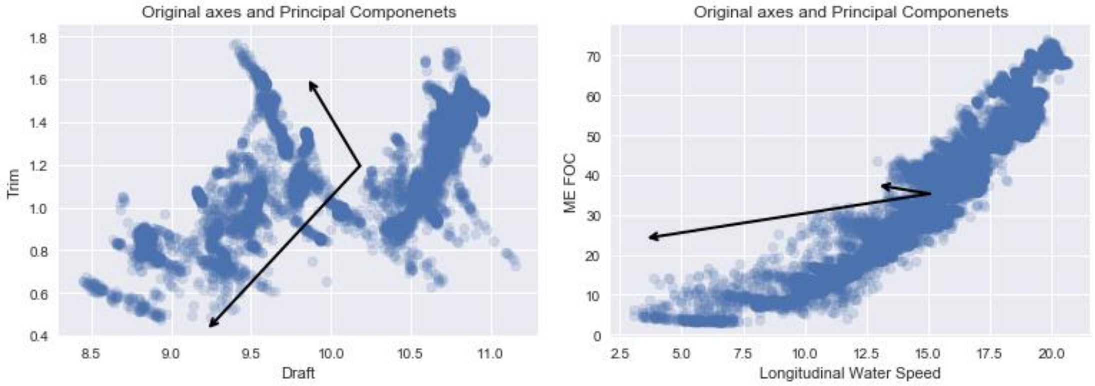


Figure 4.6 The principal components of two pairs of parameters. The arrows directions are the eigenvectors and their magnitudes are the eigenvalues of the correlation matrix of each pair of parameters.

5 Artificial Neural Networks: Implementation and Optimization

For readers who are not familiar with the concepts and the recent developments in the field of Artificial Neural Networks we recommend to read Appendix A first. It aims to introduce the fundamental concepts of Artificial Neural Networks (ANNs), to whom might be unfamiliar with, and display their mathematical formulation so that a discussion about ANNs' architecture and parameters can follow.

This chapter is targeted to the readers who have a basic familiarization with the concept of Artificial Neural Networks and are up-to-date with the recent developments in the field (after 2013). Comprehending properly the ANNs' architecture and parameters' influence on the possible outcome is of great importance for building functional and well-performing models. This argument is supported in this chapter where the results from training and testing ANN models, are presented in chronological and increasing complexity order. In summary, this chapter describes the basic tool of this study and attempts to shed light on the key-factors that fine-tune an ANN model. Also, we achieve to provide quantitative evidence on the significance of the data pre-processing and feature engineering procedures that were implemented in the previous chapters.

5.1 Parameters' Selection, Training Results and Optimization

In this section we present the results of the training process for two reference models, one that estimates the ME FOC and another the Propeller Shaft Power. Also, we investigate the effect of data pre-processing, feature engineering and some additional methods and tricks on the final performance of our models on the test sets. All the results are summarized in a matrix format and necessary details for each case are given below.

Whenever we refer to the training procedure, we mean a set of specific steps that are implemented in the following order:

a) Shuffling the data set.
b) Splitting the data set into the training set and the validation set, by default to 80% -20% except when defined otherwise.
c) Normalizing the arithmetic elements of our data, using z-score normalization (zero mean and unit variance).
d) Train a standard FNN. The FNN has the following user defined parameters:
    i. The number of layers
    ii. The number of units in each layer
    iii. The activation function of each layer
    iv. Optimizer
    v. Batch size
    vi. Error function
    vii. Number of epochs

Every one of the aforementioned steps plays a crucial role in the performance and robustness of the model. The goal is to minimize the error function of course, but this does not mean that a model with poor generalization capabilities and overfitted to the training data is desired, even if it brings the training error down to almost zero. For this reason, we will discuss briefly the importance and the benefits of each and every one of these steps in the training procedure.

Shuffling the data is crucial in order to avoid biased learning of the model. If we consider all the measured parameters in the ship dataset to be random variables, we can confidently assume that they are not static. Consequently, data collected later on time carry new information based on actual changes in our physical system, the ship. If the data are not shuffled, the model will be trained on hull conditions that describe a certain time window of the ship's operation and examples related with the later behavior of the system will be ignored by the model.

The split of the data, by convention among machine learning practitioners, varies between 70-30% and 90-10% as training-validation set. The most influential factor in the determination of the split percentages is the dataset's size. In large datasets the validation or the test set can be a smaller percentage of the whole since it is still quite large in absolute number.

Data normalization is a standard procedure for most cases (for regression models and arithmetic data) and should not be skipped. It is easier to conceptualize its importance in a two-dimensional curve fitting example. Suppose a set of data points located distantly from the axes origin of our 2-D plane and an estimation of the slope, α and intercept, β of a line that fits the data and results to the least square error. A small perturbation in the value of α is shifted to larger displacements of the line's position in points more distant regarding the axes' origin and therefor erratic behavior of the error function.

Setting the models parameters is a demanding procedure since on the one hand there are not any straight-forward technics to guide us through and on the other hand, brute force search for optimum values is inefficient. One should take under consideration a lot of theoretical background knowledge, mainly related with the subjects that were described in sections 5.1 and 5.2, but this is not enough. The reason is that selecting a relatively efficient number of layers and units or a well-suited activation functions is plausible with some experience and following basic rules of thumb, offered by the machine learning community. However, fine-tuning the model and being confident that we are verging to the global minimum of the error function is far more difficult. The strategy followed in this study can be summarized in the following propositions:

- Add layers until the error stabilizes, then add one more and if there is not improvement keep the least number of layers that achieves this performance.
- Similarly, add units to each layer until the error stabilizes, then add even more and if there is not improvement keep the least number of layers that achieves this performance.
- Experiment with fundamentally different types of activation functions. Detect the most suitable family of activation functions (i.e. tanh and sigmoid behave similarly) and test a subset of them to ensure that they do not induce deviations in the model's performance.
- Solid understanding of the training process and the available optimizing algorithms can assist immensely in the direct selection of the proper optimizer.
- The batch size is the number of training samples that are forward propagated through the network before re-adjusting the weights. If the batch size is set equal to 1 then we have a sequential model and weights are adjusted for the error estimated on each sample point. If the batch size is set equal to the total number of training samples then we have a batch-method and the weights are adjusted only once in every epoch. Other cases of intermediate batch size are mini-batch methods. We decided to couple the selection of batch size number with the number of epochs.
- The number of epochs is selected in relation to the batch size. Increasing the number of epochs means increasing the number of times that the whole training set is propagated through the model but the batch size determines the number of weight-updates that occur in each epoch. We consider the total number of weight-updates to be the effective degree of the model's training. For instance, if the number of epochs is constant, doubling the batch size leads to the half weight-updates and this is proportional to the computational time required for the model's

training. Fewer weight-updates lead to faster training but a larger batch size means less focus on the fitting of each individual data point and vice versa. Finally, it should be noted that a large number of epochs is usually responsible for overfitting the model while a large batch size can assist the model to fit better on noisy data.

- The error function in regression models is commonly selected to be the Root-Mean-Square-Error (RMSE) function and we stick to that.

In Table 5.1 we present the results of an extended training and testing procedure. The 'data set' column has the labels of differently processed datasets in an ascending order of pre-processing degree. The 'raw data' are completely unprocessed and have only been synchronized into per-one-minute timestamps. All the datasets originate from the 'raw data' and are furtherly processed.

- The 'Smoothed' (s) dataset is produced by averaging the 'raw data' in 5-minute time windows.
- The 'Smoothed & Corrected' (scor) dataset is averaged in the same way and preprocessed by the correction algorithm presented in Section 3.2.
- The 'Smoothed & Cleaned' (scl) dataset is averaged in the same way and preprocessed by the outlier detection algorithm presented in Section 3.3.
- The 'Smoothed & Corrected & Cleaned' (scc) dataset is averaged in the same way and preprocessed by both the previous algorithms.
- The 'SCC & k-folds' is just the scc dataset in which the k-fold technic for cross-validation of the generated models has been applied.
- The 'SCC Atlantic trips' originates from the scc dataset but includes only the data points that were collected while crossing the Atlantic Ocean.
- The 'SCC-fouling' is the scc dataset without the 'Fouling' feature.

The other columns in Table 5.1 present error metrics for the target of the models. One model is feature engineered and trained to estimate the Fuel Oil Consumption of the Main Engine ('ME FOC') and a second one estimates the propeller shaft power ('Shaft Power').

- The 'RMSE' column is the root-mean-squared error between the model's estimations and the target values of the validation set. If y_i is the target value and $\widehat{y_i}$ is the estimated, we compute the RMSE as:

$$RMSError = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{n}}$$

- The 'R^2-value' is the linear correlation coefficient among the estimated values and the targets values.

- The '% accuracy with respect to (w.r.t.) the mean' is defined as (Petersen, et al., 2011):

$$\frac{M_{VS} - RMSE}{M_{VS}} \times 100$$

where M_{VS} is the mean value of the target variable at the validation set.

- The 'Relative error' (Pedersen & Larsen, 2009) is computed as:

$$Relative\ error = \frac{1}{n}\sum_{i=1}^{n}\frac{|\widehat{y_i} - y_i|}{|y_i|} \times 100$$

All the models share the same number of hidden layers and hidden units, the same activation functions, batch size, epochs and the same error function. This is done in order to benchmark the pre-processing algorithms that were used for correcting or cleaning the data and to observe the effect of some features on the model's performance.

The results in Table 5.1 are the average values of five training-testing procedures in order to reduce the randomness of the outcome error metric value. The values of the error metrics support the positive effect of the correcting and cleaning algorithms in both target cases. The RMSE is drastically reduced when we use the smoothed data set and it keeps shrinking when the data have been corrected or cleaned. Accuracy improves from 92% to almost 98% when in the case of ME FOC model and from 96% to 98.5% in the case of Shaft Power. The optimum results are obtained when k-fold cross validation technic is applied and that was expected since it is a common technic in plenty machine learning applications.

Furthermore, in the scenario where data points only from Atlantic Ocean sailing are included in the dataset, the errors are even smaller because the modeled phenomenon is in a much steadier state and effects of swallow or confined waters are absent. Also, transient engine operation is much more frequent during coastal navigation rather than during Atlantic Ocean sailing and so intense nonlinearities from the engine's signals are absent as well.

Finally, a test for the effect of the 'Fouling' feature, that was generated from the dataset's temporal information (see Section 4.1), is implemented. We see that the errors are increasing when the 'Fouling' feature is omitted from the input data set and that gives us confident to utilize this generated feature in future applications.

In Figures 5.5 and 5.6 the results of table 5.1 are presented graphically and the improvement on both models' performance can be perceived immediately.

Table 5.1 Summary table of the testing results.

DATA SET	RMSE		R^2-value		Accuracy w.r.t. the mean (%)		Relative error (%)	
	ME FOC (tn/24hr)	Shaft Power (kW)	ME FOC	Shaft Power	ME FOC	Shaft Power	ME FOC	Shaft Power
Raw data	3.409	372.380	0.967	0.992	92.320	95.889	4.805	3.509
Smoothed (s)	1.802	280.081	0.991	0.995	95.940	96.905	3.613	3.066
Smoothed & Corrected (scor)	1.085	232.615	0.997	0.997	97.540	97.430	2.657	2.935
Smoothed & Cleaned (scl)	1.070	186.440	0.997	0.998	97.584	97.951	2.423	2.310
Smoothed & Corrected & Cleaned (scc)	0.913	165.665	0.997	0.998	97.837	98.180	2.423	1.873
SCC & k-folds	0.811	133.661	0.998	0.999	98.125	98.492	1.919	1.780
SCC Atlantic trips	0.648	105.122	0.998	0.999	98.747	99.025	1.107	0.848
SSC-fouling	0.947	204.670	0.997	0.997	97.863	97.751	2.162	2.308

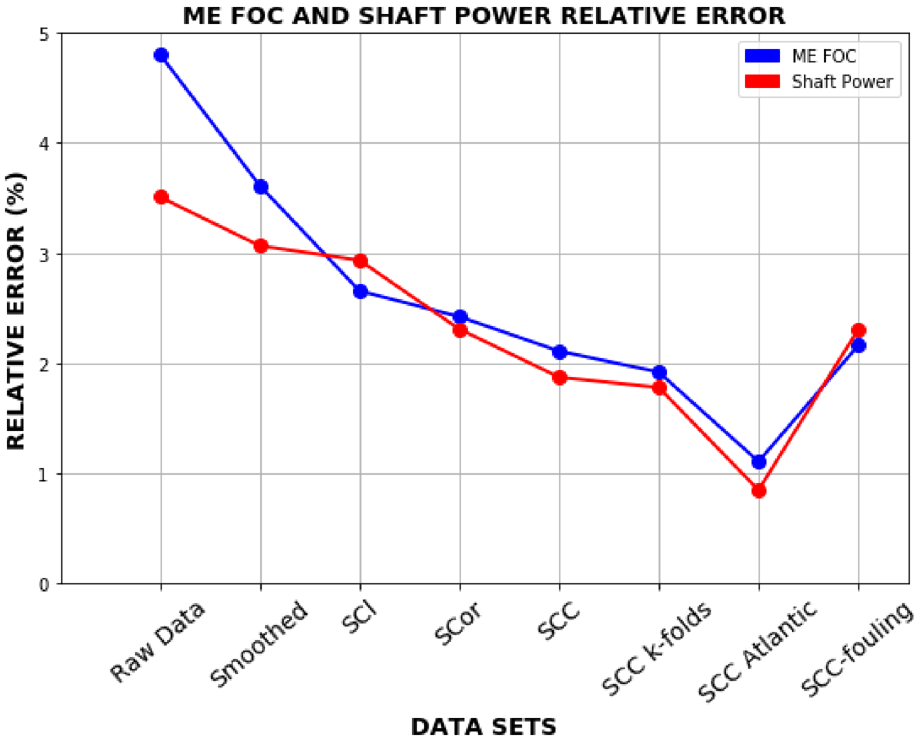


Figure 5.1 The relative error on the validation set. Each dataset on the x-axis trains a new model and relative error among the estimated and the target values is the y-axis value.

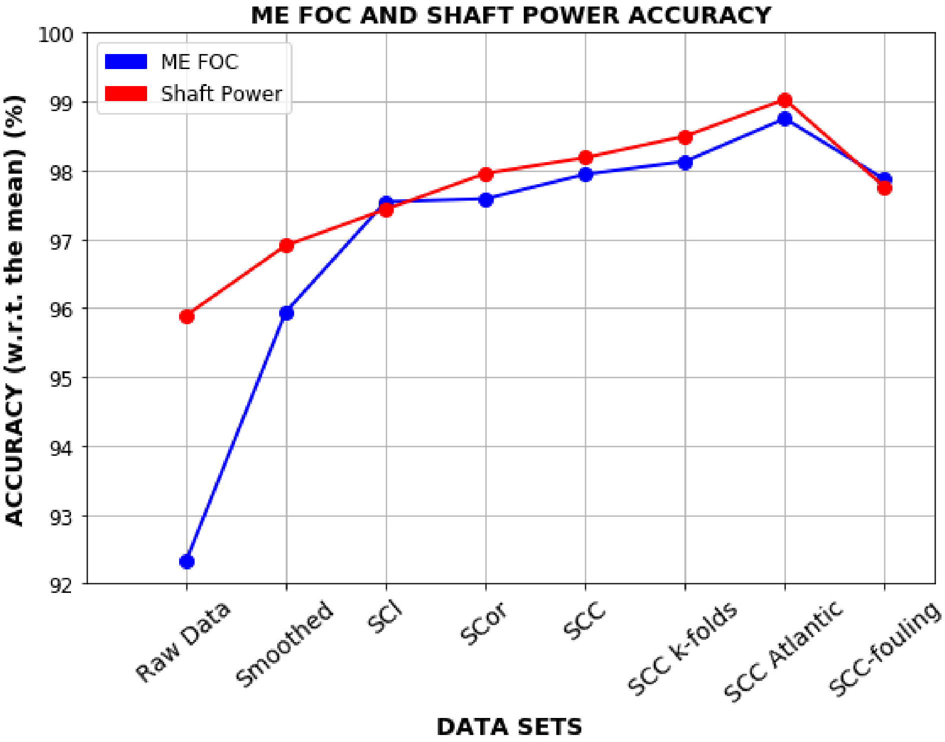


Figure 5.2 The accuracy of the models' estimations on the validation set. Each dataset on the x-axis trains a new model and the resulting accuracy among the estimated and the target values is the y-axis value.

6 Applications of ANN models on ship propulsion problems

The last chapter of this study presents three different types of applications on the field of ship propulsion modeling. We investigate how it is possible to exploit ANN models, that are built purely from the ship's operational data, and address important issues related to optimum ship performance and emissions mitigation. The models described in Section 5.3, and others similar to them, can form the core of decision support tools that assist a technical or operations department to achieve reduction of the fuel oil consumption or timely hull cleaning and effective condition monitoring. All that would lead to more energy efficient and profitable ship operations.

6.1 Application I: Prediction of the main engine's total fuel oil consumption

This first application aims to estimate the actual fuel oil consumption (ME FOC) that the ship's main engine would have during a user-defined voyage. Given that all the values for the input parameters (Table 6.1) of the model are known, it estimates the total mass of consumed fuel with an accuracy around 98.5% and above. It is a high-fidelity model that could explore the effects of speed optimization on the emission mitigation mission or provide reliable estimates on particular specifications of a charter party agreement.

Table 6.1 Input features for the ME FOC model.

Longitudinal Water Speed	STD_Longitudinal Water Speed	Draft Tot	Trim	Rudder Angle	Propeller shaft power
M/E START AIR PRESS	Wave Height	Fouling	Wind Effect		

From the available dataset of the examined ship, we chose to work with the datapoints that were collected during the transatlantic trips of the ship. In a period of 12 months we detect 12 crosses of the Atlantic by the ship and split them to 8 trips as training set, 2 trips as validation set and 2 trips as test set (Figure 6.1). We need a validation set because the k-folds technic is used for improved performance (see Section 5.3). Also, we introduce here the *ensembles* technic, which is the training of multiple identical models on the exact same data in order to average the outputs of all these models when trained, and use this value as the final estimation. Hence, when we refer to the model's estimation, we mean the average value of the 10 models' outputs.

This application provides both, the time series of the fuel oil consumption and the total consumed mass of the fuel. On every estimation, the 99% confidence interval is calculated in order to justify the term 'high-fidelity' that we mentioned before. The results from the test set's estimations are presented in Table 6.2 and in Figures 6.2. The y-axis scale in Figure 6.2 is not allowing us to display the confidence intervals around the estimation curve. That is why in Figure 6.3 a specific area of interest from the test set is presented. We should notice there that for low values of the ME FOC the uncertainty in the model increases because there are few training data on this range but overall it behaves quite robustly.

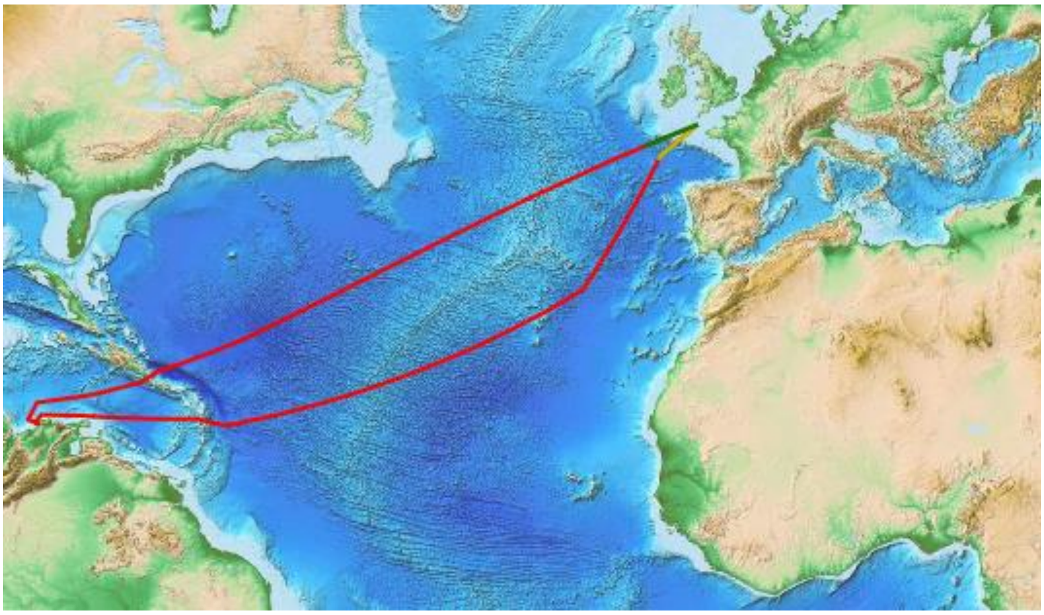


Figure 6.1 The ship's path in the test set's data points. The transatlantic trip begins from the European side (yellow part of the curve) and ends back there (green part of the curve). Its duration was about 23 days.

The error related to the estimated value of the total ME FOC in Table 6.2 is ±4.2 tons. It is calculated by propagating the standard error of each estimated value. We see that the actual value of the total ME FOC lies well within the anticipated interval. Estimations of the total ME FOC from this model could be taken into account when underlying a charter party agreement.

Table 6.2 Summary of results of the ME FOC estimation application.

DATA SET	**RMSE** (ton/24hr)	**R^2-value**	**accuracy w.r.t. the mean** (%)	**Relative error** (%)	**Total ME FOC** (ton)
SCC Atlantic trips - Ensembles	0.589	0.998	98.862	0.954	1040.257
Actual (measured) consumption (ton)					1037.324
Difference					2.933 ton
(%)					0.283 %

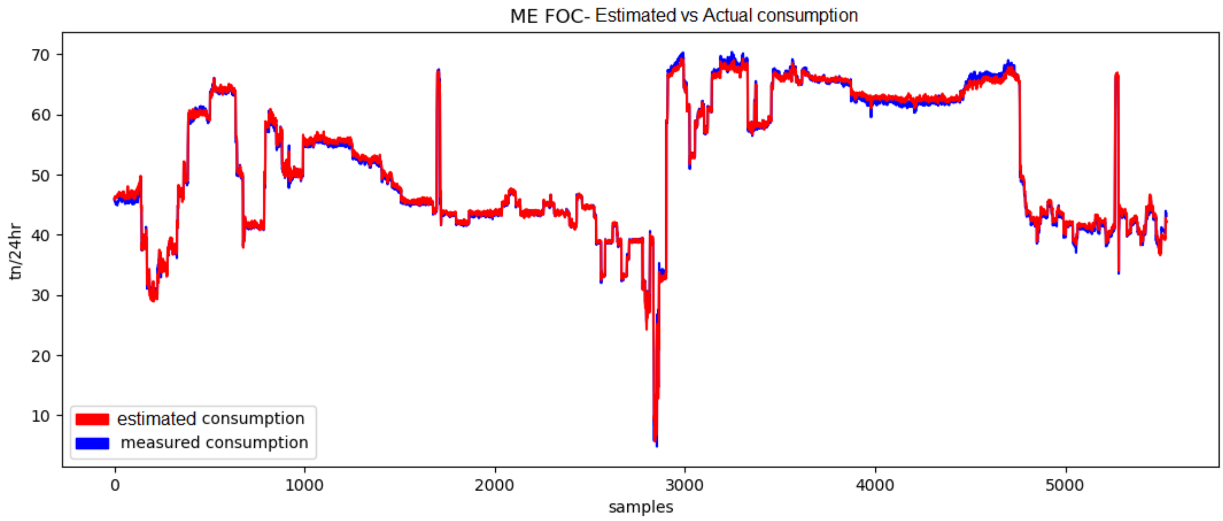


Figure 6.2 The model's estimation and the actual ME FOC on each time step (5-minute) of a transatlantic round-trip (test set).

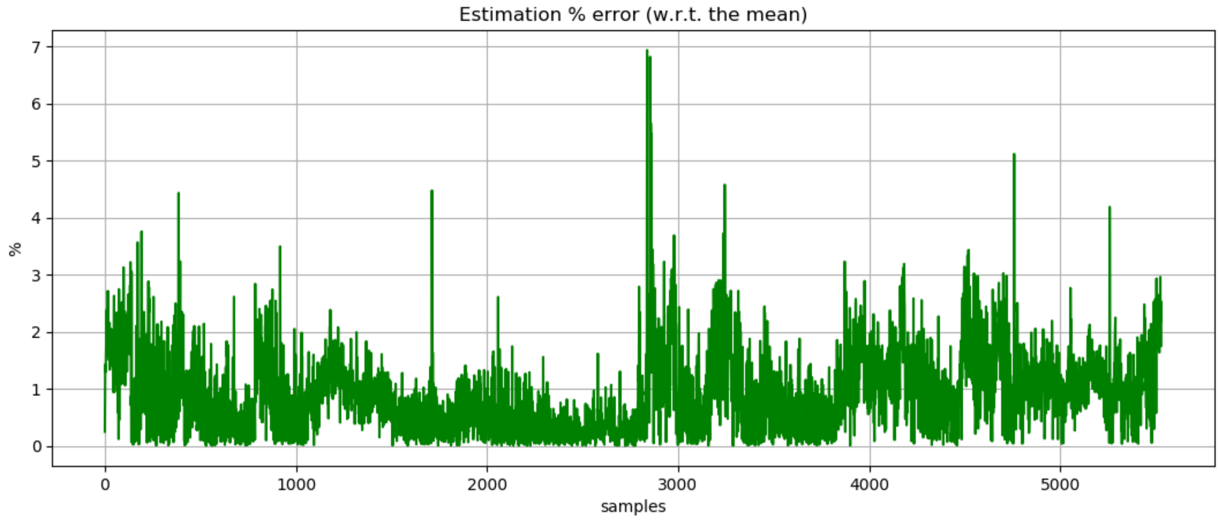


Figure 6.3 The model's estimation error as % (w.r.t.) to the mean value of FOC. The errors correspond to the estimations of Figure 6.2.

An alternative way to exploit this model is to create scenarios of reduced or increased average speed during an identical voyage. The model's output can be used to estimate the difference in the total fuel consumption, the CO_2 emissions or the fuel costs. Also, the added voyage duration can be calculated in the same, straight forward, way. The scenarios that were tested are presented in Table 6.3 along with their expected errors for the total ME FOC. In Figure 6.4 the results of table 6.3 are interpreted in terms of fuel costs and trip duration and presented as percentages in comparison to the reference scenario of the 17.75 knots. Graphs like this could provide useful information to the ship managers and support decision making.

Table 6.3 The total ME FOC for the various scenarios of reduced or increased average speed during voyage.

Average Speed (knots)	**Total ME FOC** (tons)	**Expected Error** (tons)
16.5	837.8	±4.8
17	866.56	±5.2
17.5	939.59	±4.8
17.75	993.74	±4.4
18	1124.44	±3.9

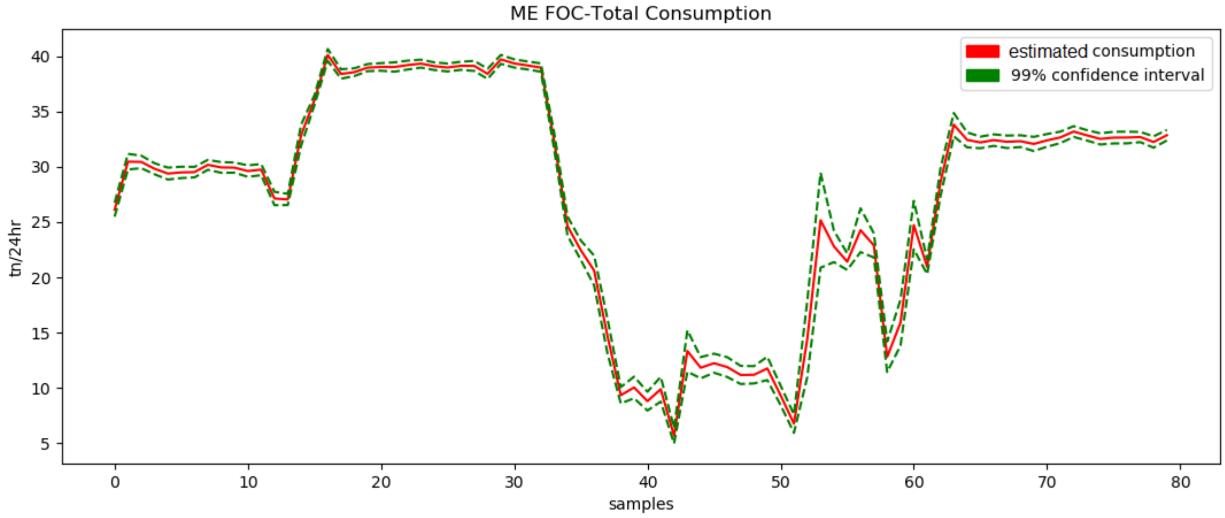


Figure 6.4 The estimated ME FOC value and the 99% confidence interval surrounding it.

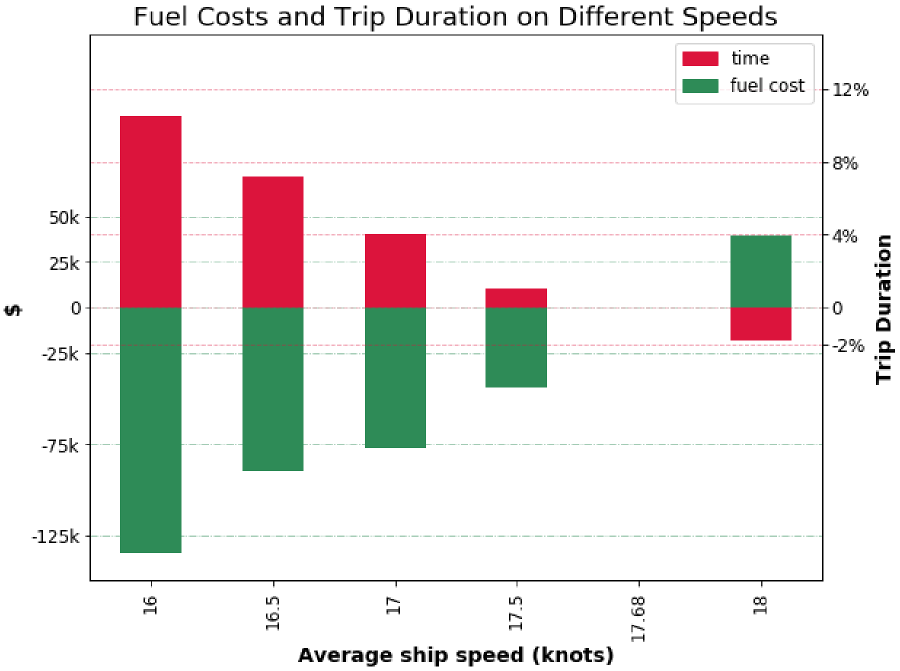


Figure 6.5 Estimated fuel costs and trip duration with respect to the reference scenario of average speed 17.75 knots.

6.2 Application II: Average Speed Loss estimation

This section presents the second application, with which we aim to offer an alternative approach to the standard ISO 19030 process for estimating a ship's average speed loss. The analysis is focused on the normalization step of the available data points. The ISO procedure for normalization is compared to an innovative way of normalizing the data via a machine learning model.

However, before normalizing the data, the ISO 19030 procedure requires strict filtering of the data in order to avoid normalizing from widely different conditions. The Table 6.4 presents the filtering requirements according to ISO 19030 and compares them with the case that an ANN model is to be used for the normalization of the data. The main difference among these two methods is that ISO 19030 requires filtering for calm sea conditions (true wind speed < 7.9 m/s) and the same loading condition. When we use an ANN model to estimates the ship's STW for any loading and weather conditions we can keep in the dataset a large number of datapoints that otherwise would be rejected. We still have to filter for large rudder angles and strong sea currents cause in these cases the physic of the problem differs fundamentally (completely different flow of the water around the hull). The standard deviation of the STW (STD_STW) is added in the filtering procedure because we wish to reject datapoints from transient or high uncertainty moments. Also, the propeller shaft power is required to be more than 9,000 kW only because the available sea trials are for a power range over this value. Finally, through the constrains in the longitude coordinate we filter for transatlantic trips only because there are no available data for water depth and temperatures. Coastal sailing datapoints will definitely include cases of swallow or confined water sailing and maneuvering close to ports or many low load and transient operations.

Table 6.4 Comparative table of ISO 19030 filtering requirements versus the proposed method.

PARAMETER		ISO 19030	ANN
Wind Speed	< 7.9 m/s	✓	
Trim	+/- 0.2% L_{BP} ref. trim	✓	
Draft	→ ±5% ref. displacement	✓	
Rudder Angle	> -5° and <5°	✓	✓
Sea Current	< 1 m/s	✓	✓
STD STW	< 0.5 knots/5-min	✓	✓
Propeller Shaft Power	>9000 kW	✓	✓
Longitude	>-79 and <1	✓	✓

After the filtering of the data, the normalization procedure allows us to make possible the direct comparison of measured data points to the expected ones in the "reference condition". Reference condition could be any loading condition of the ship for which we have the Speed-Power curve in clean hull and calm weather conditions, and hence the expected values of speed (STW) for a particular value of power. Instead of reference condition we may use a reference period which is a 3-month period (or so) of data collection after a hull cleaning. By subtracting from the expected STW values the normalized measured values of STW, for the same engine power, we get an indication of the ship's performance in the form of a speed loss (see Equation 6.1).

$$V_d = 100 \cdot \frac{V_m - V_e}{V_e} \qquad (6.1)$$

where V_e is the expected value of STW (in reference condition) and V_m is the measured value of ST (Petersen & Winther, Mining of Ship Operation Data for Energy Conservation, 2011)W. If V_m is not measured in the exact same displacement as the reference condition, it should be normalized according to Equation 6.2.

$$V_2 = V_1\left(\frac{\Delta_1^{2/3}}{\Delta_2^{2/3}}\right)^{1/3} \qquad (6.2)$$

Where $V_2 (= V_m)$ is the speed at reference condition (displacement Δ_2), V_1 is the speed at measured displacement Δ_1. After the calculation of the V_d values for all the available data points n, we take their mean and this results to the *average speed loss* $\overline{V_d}$ (see Equation 6.3).

$$\overline{V_d} = \frac{1}{n}\sum_i^n V_{d,i} \qquad (6.3)$$

In Figure 6.5 we have estimated the average speed loss (ASL) with both methods. The blue dots are calculated through the ISO 19030 normalization formula (Equation 6.1) while the red dots are obtained by normalizing the speed values with the ANN model. Obviously, in the case of the ANN we have many more available data points to support our estimation because the filtering procedure is not so strict. Also, both methods estimate the almost the same value for the ASL but the ANN methods trends correctly (increasing speed loss over time) while the ISO 19030 method estimates a decreasing speed loss over time. We believe that this happens mainly due to the limited number of data points involved in the estimation of the ASL. The ASL value estimated from the ANN model is 5.12% and from the ISO 19030 is 5.99% while the available data points after filtering are about 22,000 and 2,500 respectively.

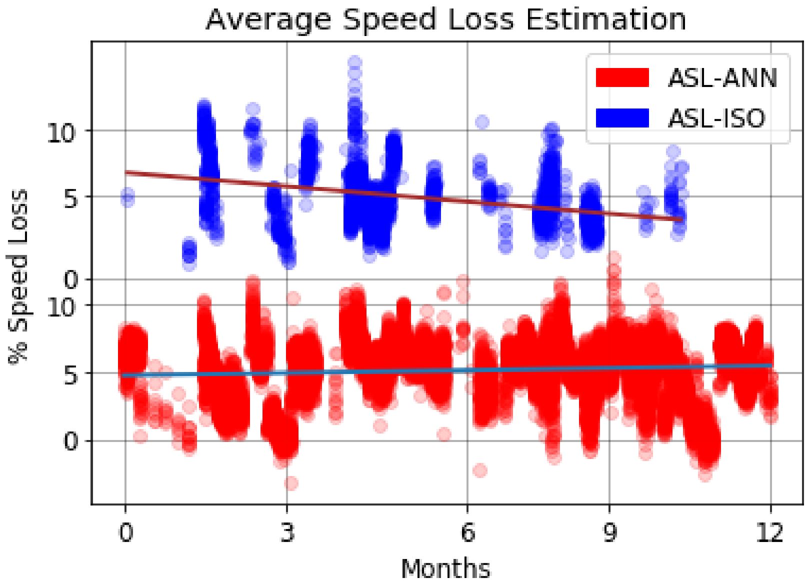


Figure 6.6 Estimation of the ship's average speed loss via two different normalization methods.

7. Conclusions and Recommendations

This study attempted to investigate the margins of improvement in the data-driven ship propulsion models, also known as black box models, for the scopes of performance analysis and emissions reduction. We choose to build our models with multi-layer Artificial Neural Networks (ANNs) and therefore, two important stages in this modeling procedure were the data pre-processing stage and the neural network's parameters' (and optimizer) selection stage. These stages were investigated thoroughly and proved to be able to contribute remarkably to the high accuracy that was achieved by the models.

More specifically, in the pre-processing stage, two algorithms for correcting and cleaning the data were proposed, and when applied to the available dataset they increased the accuracy of the produced models by ~1.5%. However, the initial accuracy of the ANN models was quite high, from 93% to 95%, depending on the targeted parameter (FOC, Speed or Power). This high initial accuracy was achieved due to the much larger number of hidden units that were used in our networks, in comparison to other works (Senteris, 2018), (Besikci, et al., 2015) and (Pedersen and Larsen, 2009).

Finally, in the applications chapter, we demonstrated how a model that predicts the ship's fuel oil consumption can be utilized for examining scenarios of different service speeds for the ship. Also, we implemented the average speed loss estimation according to ISO 19030 procedure and then attempted to alter the filtering and normalization by utilizing again a neural network model, that could predict the expected ship's speed based on the shaft power value, for the reference conditions. The results were quite rational and encouraging for further investigation.

Overall, the following remarks and recommendations were drawn from this study.

i. For a systematical approach to the issue of data-driven propulsion modeling, data quality metrics should be introduced. Data correction and cleaning methods should be established and applied when poor data quality is detected. In this way, it would be possible to provide datasets of similar quality in the machine learning algorithms and benchmark the various data-driven models. In 2011 Petersen had stated the need for benchmarking the ship propulsion models, in his PhD thesis (Petersen & Winther, 2011).

ii. Two abstract stages in the data-driven modeling procedure are the visual data inspection stage and the feature engineering stage. They demand relative experience on the ship's operational patterns and good knowledge of ship theory. Also, strong intuition on the physical side of the problem as well as in the data manipulation issue can be regarded as an asset.

iii. Ultimately, a deep understanding of how the data, that are constructing the model, interact with it and what are the capabilities and limitations of the produced model is essential for the proper treatment and utilization of the model, later.

iv. Last but not least, peak performance is noticed when the open sea (Atlantic) data points are isolated and used to build the model. Probably, that happens because there is missing information for the other areas that the ship sails. A more extended dataset could offer new and meaningful features for the model and therefore increase its accuracy, among the different areas that the ship operates. It could include the following additional parameters: Water depth, Seawater, and Air temperature, Salinity, Humidity, Swell significant wave height and direction and perhaps most importantly the control commands from the bridge (control signals for the engine, the rudder, the ballast pumps, etc.)

References and Bibliography

Aldous, L. (2015). *Ship Operational Efficiency: Performance Models and Uncertainty Analysis.* London: University College London.

Besikci, B. E., Arslan, O., Turan, O., & Olcer, A. (2015). An artificial neural network based decision support system for energy efficient ship operations.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Cambridge: Springer. ISBN-13: 978-0387-31073-2.

Bouman, E. A., Lindstad, E., Rialland, A. I., & Stromman, A. H. (2017, April 4). State-of-the-art technologies, measures, and potential for reducing GHG emissions from shipping - A review. *Transportation Research Part D*, pp. 408-421.

Chollet, F. (2015). *Keras.* https://keras.io.

Coraddu, A., Oneto, L., Baldi, F., & Anguita, D. (2016). Vessels fuel consumption forecast and trim optimisation: A data analytics perspective. *Ocean Engineering* .

DNV-GL. (2017). *Low Carbon Shipping Towards 2050.* DNV-GL.

Hasselaar, T. W. (2010). *An investigation into the development of an advanced ship performance monitoring and analysis system.* Newcastle University: School of Marine Science and Technology (PhD Thesis).

Haykin, S. (2009). *Neural Networks and learning machines.* Ontario: Pearson Education Inc. ISBN-13: 978-0-13-147139-9.

Holtrop, J., & Mennen, G. (1982). An approximative power prediction method. *International Shipbuilding Progress 29 (335)*, pp. 166-170.

Hotelling, H. (1933). Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology, 24*, pp. 417-441.

IMO. (2012). *International Convention for the Prevention of Pollution from Ships (MARPOL)/ Annex VI.* IMO.

IMO. (2015, December 14). *www.imo.org*. Retrieved from Press Briefings: http://www.imo.org/en/MediaCentre/PressBriefings/Pages/55-paris-agreement.aspx

ISO 19030. (2016). *Ships and marine technology- Measurement of changes in hull and propeller perfomance.* Geneva: International Organization of Standarization.

Karaminas, L., & Shen, T. (2016). Ship Powering Perfomance - Learning from the challenges faced by owners. *Energy Efficient Ships.* London: The Royal Institution of Naval Architects.

Kingma, P. D. (2015, July 23). ADAM: A method for stochastic optimization. *arXiv: 1412.6980v8 [cs.LG]*.

Kriesel, D. (2007). *A Brief Introduction to Neural Networks.* available at http://www.dkriesel.com.

Leifsson, L. T., Saevarsdottir, H., Sigurdsson, S. T., & Vestainsson, A. (2008). Grey-box modeling of an ocean vessel for operational optimization. *Simulation Modeling Practice and Theory 16(8)*, pp. 923-932.

Lewis, E. V. (1988). *Principles of Naval Architecture (2nd Revision), Volume II - Resistance, Propulsion and Vibration.* Society of Naval Architects and Marine Engineers (SNAME). ISBN 978-0-939773-01-5.

Logan, K. P. (2011). Using a Ship's Propeller for Hull Condition Monitoring . *ASNE Intelligent Ships Symposium IX.* Philadelphia, PA, USA.

MEPC.1/Circ.684. (2009). *Guidelines for voluntary use of the ship Energy Efficiency Operational Indicator (EEOI).* Londod: IMO.

MEPC.59. (2009). IMO.

MEPC.62. (2011). *Resolution MPEC.203(62).* IMO.

Microsoft. (2019). *Microsoft Azure*. Retrieved from https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine 2*, pp. 559-572.

Pedersen, B. P., & Larsen, J. (2009). Modeling of Ship Propulsion Performance. *World Maritime Technology Conference WMTC 2009.* Mumbai: The Institute of Marine Engineers.

Petersen, J. P., & Winther, O. (2011). *Mining of Ship Operation Data for Energy Conservation.* Kgs. Lyngby, Denmark: Technical University of Denmark (DUT). (IMM-PHD-2011; No. 264).

Petersen, J. P., Jakobsen, D. J., & Winther, O. (2011). A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions. *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11)*, (pp. 305-316). Berlin.

Psaraftis, H. N. (2019, April 15). Speed Optimization vs Speed Reduction: the Choice between Speed Limits and a Bunker Levy. *MDPI- sustainability*.

Raschka, S. (2015). *Python Machine Learning.* Packt Publishing Ltd. ISBN 978-1-78355-513-0.

Ruder, S. (2017, June 15). An overview of gradient descent optimization algorithms. *arXiv:1609.0474v2 [cs.LG].*

Senteris, A. (2018). *On the Estimation of the Propulsion Power of a VLCC Tanker Based on Operational Data.* Athens: National Technical University of Athens (Diploma Thesis) .

Smith, T., P. Jalkanen, J., A. Anderson, B., Corbett, J., Faber, J., Hanayama, S., . . . Pandey, A. (2014). Third IMO GHG Study 2014. *Executive Summary and Final Report.*

Taylor, D. W. (1910). *The Speed and Power of Ships.* New York: Society of Naval Architects and Marine Engineers.

Themelis, N., Spandonidis, C. C., & Giordamlis, C. (2018a). Data acquisition and processing techniques for a novel Perfomance Monitoring System based on KPIs. *IMAM.*

Themelis, N., Spandonidis, C. C., Christopoulos, G., & Giordamlis, C. (2018b). A comparative study on Ship Performance Assessment based on Noon Report and Continuous Monitoring datasets. *12th Conf. Hellenic Institute of Marine Technology*, (pp. 55-64). Athens.

Theotokatos, G. P. (2007). A Modelling Approach for the Overall Ship Propulsion Plant Simulation. *6th WSEAS International Conference on SYSTEM SCIENCE and SIMULATION in ENGINEERING*, (pp. 80-87). Venice, Italy.

Wartsila . (2015, June). *www.wartsila.com*. Retrieved from Encyclopedia-2nd edition: https://www.wartsila.com/encyclopedia/term/speed-logs

Wikipedia. (2019, May 28). *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Feature_engineering

Appendix I- Neural Networks

History and Mathematical Formulation

The history of Artificial Neural Networks is actually the history of the human endeavors to design intelligence machines. Once the miraculous invention of the electronic computer was established, some scientists' attention focused on the inner function of the human brain.

The firsts to inform us about the neurons in our brain were Warren S. McCulloch, a neuroscientist, and Walter Pitts, a logician. In 1943 they published their work in the Bulletin of Mathematical Biophysics with the title "A logical calculus of the ideas immanent in nervous activity". In their paper, they describe how networks of neurons in the human brain may work and provide a simple model for the single neuron (see Figure 5.1). This model, when used as the building block of the computational analogous of the biological neural network in the brain, had the ability to approximate almost any arithmetic or logic function.

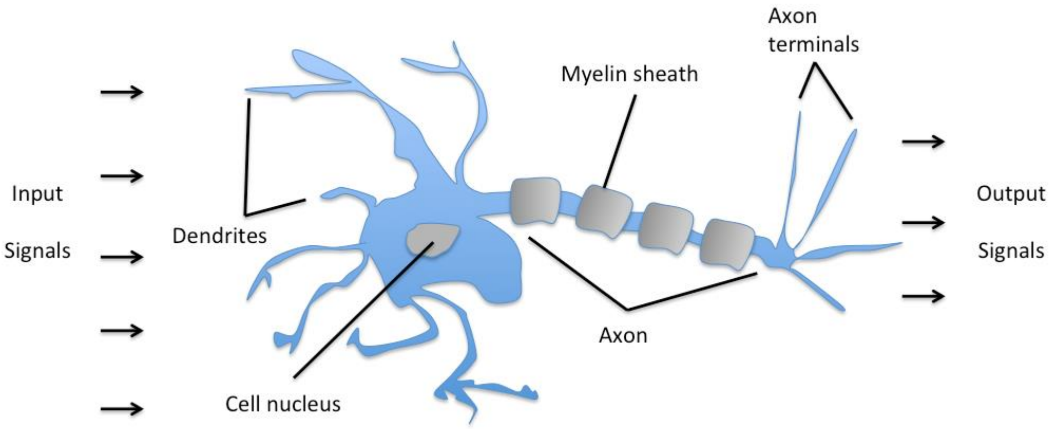


Figure 0.1 An illustration of a biological neuron with some of its biological features that relate to the model of perceptron, introduced by McCulloch and Pitts and later formulated by Rosenblatt. (Source: Raschka,2015)

The work of McCulloch and Pitts inspired many other researchers to expand the idea of the neuron and develop the first applications as well as the mathematical foundations for the Artificial Neural Networks. In 1951 Marvin Minsky, as a student built a neurocomputer but did not manage to come up with any application for it. In contrast, Frank Rosenblatt in 1958 at the Cornell Aeronautical Laboratory created the first successful neurocomputer, a machine that was able to see and classify linearly separable objects. Even though Rosenblatt presented the perceptron as a machine, it was the algorithm that simulates the operation of a neuron that laid the foundations for the development of the modern ANNs.

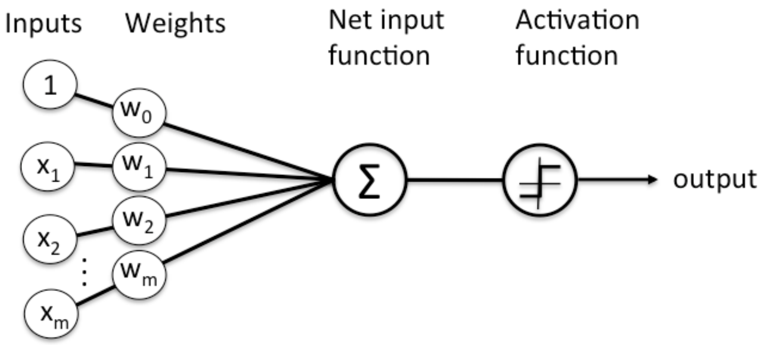


Figure 0.2 Schematic of Rosenblatt's perceptron. The basic elements consisting the perceptron's model are the inputs, the weights, the net input and the activation function and the output. Until today they have not change. Source: *(Raschka, 2015)*

Even though the fundamental model of the perceptron existed since 1958, scientist keep pushing forward the research and development on the topic only for the next ten years. After that, the research halted, mainly because of Minsky's and Papert's book, Perceptron (1969), and we entered the so-called AI winter. Only in the mid 80's scientific progress on the field initiated again.

However, before catching up with the most recent developments in the field of ANNs we shall present the basic mathematical formulation and explain few details of what we have already discuss. In Figure 5.1 the illustration of the neuron, as perceived by McCulloch and Pitts, even though is simple, includes a certain amount of biological information that is not essential for a computational model of the neuron, the Rosenblatt's perceptron, presented in Figure 5.2.

In simple words, the computational neuron of Figure 5.2 represents the electric impulses, that a biological neuron senses in its *Dendrites* (see Figure 5.1), as scalar numbers. Then the *Cell nucleus* and the *Axon* that combine and transmit the input signals towards the outputs are modelled as a linear combination of the scalar inputs, its one receiving an individual weight, and then their sum passes through an activation function. This activation function, originally a step function, simulates the firing or not of the *Axon terminals*.

The mathematical formulation of Rosenblatt's perceptron is the elegant equation (5.1).

$$y(x, w) = h\left(\sum_{j=1}^{M} w_j x_j\right) \qquad (5.1)$$

Where $h(\,)$, is a nonlinear activation function in the case of classification, like the image recognition problem that Rosenblatt attempted to solve with the step activation function. In the case of regression is the identity function. The x and w denote the input and weight vector, $(x_1, x_2, \ldots, x_m)$and $(w_1, w_2, \ldots, w_m)$ respectively, and y is the perceptron's output.

At this point, additional terminology that is used in the field of Neural Networks can be introduced. Equation (5.1) describes a single layer, single output ANN with m-inputs. The weights, w are the adaptive parameters that are to be learnt by the ANN in the training phase, so as to map correctly the input vector x to an output value y. The training procedure of the ANN is analytically described in a following section. For the moment we mention only that for every available pair of data (x_n,t_n), the weights are adapted in order to have the output y_n, that corresponds to each input x_n, as close as possible to the known *target value* t_n. Now, if another perceptron is placed at the output of the first, so as the value $y_n$becomes its input, a two-layer ANN is created. Alternatively, a two-layer ANN is sometimes called a single-hidden layer network.

A more generic formulation of the Rosenblatt's perceptron can be used to model a Neural Network. Placing perceptrons in parallel increases the width of the network and placing them in series increases the depth (hidden layers) of the network. If we examine the example of a two-layer ANN, for the outputs of the first layer in the network we may write

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \qquad (5.2)$$

Where $j = 1,2,\dots,M$ corresponds to the number of "in-parallel" perceptrons and the superscript (1) to the first layer's parameters. Accordingly, the $w_{j0}^{(1)}$ are the *biases* that are added to each linear combination of inputs, on each layer. The resulting values a_j are known as *activations*. Each of them is then transformed using a differentiable, nonlinear activation function $h(\,)$ to give

$$z_j = h(a_j) \qquad (5.3)$$

Consecutively, these values are passed to the next layer as inputs and are called *hidden units*. In the context of linear algebra, the z_j form the *basis functions* of the model's space since it is their linear combination that produces the final value in the output. In the second layer of the network the z_j are again linearly combined to give *output unit activations* and so on and so forth till the last layer of the network that gives the output of the model.

In our example we get

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} x_j + w_{k0}^{(2)} \qquad (5.4)$$

where$k = 1,2,\dots,K$, and K is the total number of outputs. This transformation corresponds to the second layer of the network, and again the $w_{k0}^{(2)}$ are bias parameters. Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs y_k. The choice of activation function is determined by the nature of the data and the assumed distribution of target variables. The most commonly used activation functions will be discussed in a following section.

With the combinations of Equations (5.2), (5.3) and (5.4) we get the overall network function

$$y_k(x,w) = h\left[\sum_{j=0}^{M} w_{kj}^{(2)}\, h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)\right] \qquad (5.5)$$

where the set of all weight and bias parameters have been grouped together into a vector **w**. Thus, the neural network model is simply a nonlinear function from a set of input variables $\{x_i\}$ to a set of output variables $\{y_k\}$controlled by a vector **w** of adjustable parameters. The bias parameters in (5.5) have been absorbed into the set of weight parameters by defining an additional input variable x_0 whose value is clamped at x_0 = 1.

The model of the two-layer ANN is illustrated in Figure 5.3 with the help of a network diagram, as found in (Bishop, 2006).

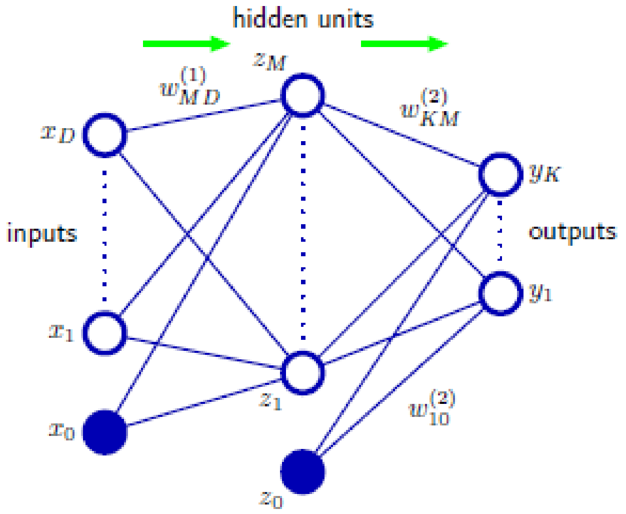


Figure 0.3 Network diagram for the two-layer neural network corresponding to (5.5). The input, hidden, and output variables are represented by nodes, and the weight parameters are represented by links between the nodes, in which the bias parameters are denoted by links coming from additional input and hidden variables x$_0$ and z$_0$. Arrows denote the direction of information flow through the network during forward propagation. Source: *(Bishop, 2006)*

Finally, an important piece of information that should be pointed out are the approximation properties of ANNs. Studies of many scientists have found that feed-forward networks, like the neural network described in Figure 5.3, could be characterized as *universal approximators*. The impact of this property is better understood by an example that we find in (Bishop, 2008): “a two-layer network with linear outputs can uniformly approximate any continuous function on a compact input domain to arbitrary accuracy provided the network has a sufficiently large number of hidden units. This result holds for a wide range of hidden unit activation functions, but excluding polynomials.” Of course, in practice is not so easy to find suitable values for the parameters that would lead to zero approximation error. This challenge is primarily related to the training of the neural networks and its architecture, as we will discuss in the following section.

Architecture and Training of Feed-forward Neural Networks

In the present section, the **most common architectures** of ANNs that are encountered in the literature are briefly described. The goal is to generalize further the example that was used to introduce the basic principles and formulation of ANNs, leading to Equation (5.5) and Figure 5.3, in order to explore their capabilities.

For instance, the presented two-layer network with M-hidden units allows us to imagine how a more complex ANN would be like. Instead of having two layers one can choose to have an arbitrary number of N layers that are fully connected to the previous layers and every layer has an arbitrary number of M hidden units. Fully connected means that every hidden unit's output turns to an input to every hidden unit of the next layer. With $M_i, (i = 1,2, \dots, N)$ are denoted the number of hidden units in every layer of the network and so when describing the layout of the network we may write

$$[D - M_2 - \cdots - M_N - K]$$

where $D = M_1$ is the dimension of the input vector and K is the dimension of the output vector. A network with 10 inputs, two fully connected hidden layers with 50 hidden units each and 2 outputs is denoted [10-50-50-2].

In the cases where a neural network functions the way we described so far, by forward propagating linear combinations of the input values to pass through an activation function and these activations become the inputs for the next hidden layer, it shall be called Feed-forward Neural Network (FNN). Frequently FNNs with more than one hidden layer are called Deep FFNs but in this study we shall make no distinctions between different types of FNNs.

However, there are variate types of ANNs, that differ fundamentally in some parts from the simple FNNs. Two popular types of ANNs that deal well with time-series data (acoustics, natural language, video etc.) are the Recurrent Neural Networks (RNN) and the Long-Short Term Memory Networks (LSTM). The first, introduces a new type of hidden units, called *recurrent cells*, that compute the output based on, not only on their input but on their previous outputs as well. Hence, in the hidden layers, the current output of a unit becomes its input in the next computational step and in this way a kind of "memory" is added on the network. In the latest, the feature of memory is explicitly added to the hidden units by storing a certain number of previous information and regulating which of them will pass to the next layer and which will be deleted. The regulation is executed by some new structures in the network called *gates*.

Besides temporally sequential data, another common category of data is the high-dimensional data. An image that consist of 400x400 pixels is actually a dataset with 1,6E+10^4 dimensions or features. For many computer vision applications, dealing with this type of data is a necessity and one of the most effective and popular tools so far have been the Convolutional Neural Networks (CNN). This type of neural network has built-in invariance properties and so they manage to create models that are invariant to certain transformations of the inputs. In image recognition application we require from the model to classify many different transformations of an object on the same class, since the object remains the same but it could have undergone a translation and a rotation transformation, plus some scaling, probably.

The most distinguishing characteristic of CNNs is that they feature *convolution* units (or cells) in their first layers where they actually perform the equivalent of the mathematical operation of convolution onto a batch of input data and that is why they are called pooling layers. They commonly employ many hidden layers and they achieve to perform automatic feature extraction by propagating forward specific patterns of the data. Their last layers before the output are identical to an FNN.

The literature contains many more types of ANNs and probably new types will keep emerging, but for the scope of this study and in accordance with the present trends, the basic ones have been described above. Additionally, the training procedure of an ANN model on the available data is of immense importance. Even if a proper type of model is selected, the training algorithm and the optimization of hyperparameters' values are the elements that could lead us to peak performance.

The process of training an ANN refers to the estimation of the adaptive parameters' values that lead to the minimum model error. Summarizing what we have already discuss, a neural network could be perceived as parametric nonlinear function that given a set of input vectors $\{x_n\}$, where $n = 1,2, \dots, N$, together with a corresponding set of target vectors $\{t_n\}$, learns to approximate with its outputs $\{y_n\}$ the target vectors. Obviously, we wish this approximation to be as accurate as possible and this can be expressed in mathematical form with an error function. Without loss of generality, we define an objective function that consists of the sum-of-square error function, and minimize it with respect to the adaptive parameters vector w,

$$E(w) = \tfrac{1}{2}\textstyle\sum_{n=1}^{N}\|y(x_n, w) - t_n\|^2. \qquad (5.6)$$

In the case of a regression problem with a single target variable $t \in R$, we provide a probabilistic interpretation to the network outputs, y. We may also assume that t has a Gaussian distribution with an x-dependent mean, which is given by the output of the neural network, so that

$$p(t|x, w) = N(t|y(x, w), \beta^{-1}) \qquad (5.7)$$

where β is the precision (inverse variance) of the Gaussian noise. For the conditional distribution in Equation (5.7) we assume that the network's output activation function is the identity, because such a network can approximate any continuous function from $\boldsymbol{x}$ to y. Suppose a data set of N independent and identically distributed observations $X = \{x_1, \dots, x_N\}$ and the corresponding target values $\{t = t_1, \dots, t_N\}$, we construct the likelihood function of

$$p(t|X, w, \beta)$$

as the product of the dependent probabilities of obtaining each target value from the corresponding inputs on the model

$$\prod_{n=1}^{N} p(t_n|x_n, w, \beta)$$

and by taking the negative logarithm of the likelihood function we obtain

$$\tfrac{\beta}{2}\textstyle\sum_{n=1}^{N}\{y(x_n, w) - t_n^2\}^2 - \tfrac{N}{2}ln\beta + \tfrac{N}{2}ln(2\pi) \qquad (5.8)$$

which is the familiar error function of the sum-of-squares. In order to obtain (5.8) we substituted the value of the conditional probability of t with the analytical expression of the normal distribution. It occurs that maximizing the likelihood function is equivalent to minimizing the error function in Equation (5.6).

The value of **w** found by minimizing (5.6) will be denoted **w**ML because it corresponds to the maximum likelihood solution. In practice, the nonlinearity of the network function $y(x_n, w)$ causes the error E(**w**) to be nonconvex, and so local maxima of the likelihood (or local minima of the error function) may be found.

Assuming that **w**ML is found, the value of β can be calculated from (5.8) as

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, w_{ML}) - t_n\}^2 \qquad (5.9)$$

If we have multiple target variables, and we assume that they are independent conditional on **x** and **w** with shared noise precision β, then the conditional distribution of the target values is similar to (5.7), with the only difference being that β is multiplied by a *K-by-K* unity matrix, where *K* is the number of target variables. Following the same process and the same assumptions we get that the noise precision is now given by

$$\frac{1}{\beta_{ML}} = \frac{1}{NK}\sum_{n=1}^{N}\|y(x_n, w_{ML}) - t_n\|^2 \ . \qquad (5.10)$$

Furthermore, we shall introduce one more essential equation that occurs from the natural pairing of the error function and output unit activation function (see Bishop,2008 p. 234). Because in the case of regression the output activation function is the identity ($a_k = y_k$) and the error function is the sum-of-squares function we have

$$\frac{dE}{dy_k} = \frac{dE}{da_k} = y_k - t_k \qquad (5.11)$$

where *k* is the corresponding output unit. Equation (5.11) will be used again when discussing the *error backpropagation* in a following paragraph.

Let us now reconsider the issue of determining a weight vector **w** that minimizes the defined error function. One can imagine the error function as a surface on the weight space, where the components of the vector **w** correspond to the main axes of the space. Finding the coordinates of local or global minima in the surface is the solution to our problem. Consider an initial position for the vector **w** and then a small step in the weight space, from **w** to **w**+δ**w**. The change in the error function could be approximated as: $\delta E \cong \delta w^T \nabla E(w)$, where the vector $\nabla E(w)$ is the gradient of the error function at this point. Because the function E(**w**) is a smooth continuous function of **w**, when we get to the point where the gradient vanishes or in other words

$$\nabla E(w) = 0$$

we have encountered a stationary point which may be a minima, maxima or saddle point. This geometrical perception of the problem is illustrated in Figure 5.4.

It is clear that by moving in the direction of $-\nabla E(w)$ the error tends to shrink and an optimal solution can be achieved, sooner or later. However, the fact that the error function has a highly nonlinear dependence on the weights and bias parameters implies severe complexity in the form of the error surface and the existence of many points in the weight space where the gradient vanishes. It can be proven from the symmetrical properties of the neural network that for any point **w** that is a local minimum there will be numerous other points in the weight space that are equivalent minima. In a two-layer FNN for every such point exist $M!\,2^M$ equivalent points. Of course, there will also be multiple alternative stationary points. Since analytical solutions for so complex functions are impossible to be found we shall exploit methods that originate from the widely studied problem of continuous nonlinear functions optimization.

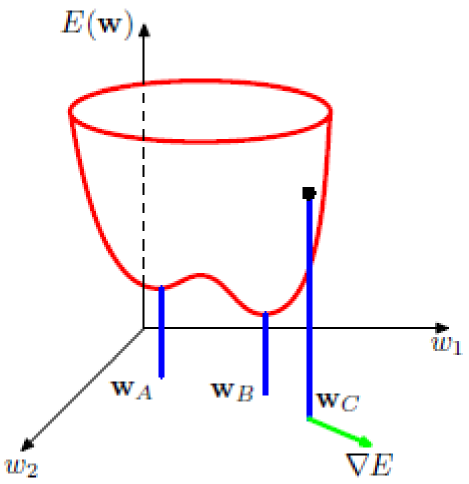


Figure 0.4 Geometrical view of the error function E(w) as a surface sitting over the weight space. Point w_A is a local minimum and w_B is the global minimum. At any point w_C, the local gradient of the error surface is given by the vector ∇E. (Source: Bishop,2008)

The most common ways of addressing such problems involve the following steps:

- Initialization of the weight vector **w** to $w^{(0)}$
- Recursive steps in the weight space according to the Equation (5.12):

$$w^{(\tau+1)} = w^{(\tau)} + \Delta w^{(\tau)} \qquad (5.12)$$

- Use of gradient information for updating the weights, meaning that the term $\Delta w^{(\tau)}$ becomes a function of $\nabla E(w)$ at the region of $w^{(\tau)}$ or $w^{(\tau+1)}$.

The idea of utilizing gradient information seems very effective and is quite popular but that does not mean that is always simple to accumulate this information. It is often computationally demanding to get a precise value for the gradient of the error function in the region around **w** and so, local approximations are used. In many cases, a Taylor expansion of $E(w)$, till the first or second order terms, can provide a satisfactory local approximation of $\nabla E(w)$. The degree of the polynomial approximation of the local gradient is classifying the optimization methods to first-order and second-order or Newton methods.

Based on the analysis so far, a few more essential elements that are utilized in the majority of the ANN training algorithms shall be introduced. For instance, many training algorithms that make use of the gradient information further modify Equation (5.12) by inserting a scaling factor η and so we have

$$w^{(\tau+1)} = w^{(\tau)} + \eta \nabla E\left(w^{(\tau)}\right). \qquad (5.13)$$

The parameter $\eta > 0$ is known as *learning rate* and is responsible for controlling the step size in the weight space. In some algorithms it has constant value while in others is an adjustable parameter in order to escape from regions with vanishing gradient, like saddle points.

Furthermore, the gradient of the error function is actually a function of the derivative of the error with respect to every weight's value, $\frac{dE}{dw_{ji}}$. By considering the definition of the derivative of a function we get the physical meaning of $\frac{dE}{dw_{ji}}$,which is how much a small change in the value of w_{ji} will affect the value of the error. But the error is a function of the output of the network itself, and the output of the network is a function of the inputs and the weights,$y_{nj} = y_j(x_n, w)$, so for a sum-of-square error function we write

$$\frac{dE_n}{dw_{ji}} = \left(y_{nj} - t_{nj}\right)x_{ji}. \qquad (5.14)$$

The Equation (5.14) links the derivative of the error function to the input and output values that correspond to the weight w_{ji}. But the output end of the weight y_{nj} is the activation a_j and the input end of the weight is the value of the activation function for the previous layer $z_i = h(a_i)$. Yet the output end is the product of the input end times the weight, so $a_j = w_{ji}z_i$ and if we differentiate with respect to the weight, we get

$$\frac{da_j}{dw_{ji}} = z_i. \qquad (5.15)$$

Also, the chain rule can be applied to the error function derivative

$$\frac{dE_n}{dw_{ji}} = \frac{dE_n}{da_j}\frac{da_j}{dw_{ji}} \qquad (5.16)$$

and if Equation (5.15) is substituted in (5.16)

$$\frac{dE_n}{dw_{ji}} = \delta_j z_i \qquad (5.17)$$

where $\delta_j = \frac{dE_n}{da_j}$, which is just a new simple notation for the derivative of the error function with respect to the output activation value. Now for the output unit *k*, bring in mind that the network's output the activation function is the identity and the error function is the sum-of-squares, hence

$$\delta_k = y_k - t_k. \qquad (5.18)$$

For the arbitrary hidden layer, the δ_j can be estimated based on Equation (5.18) and the chain rule again, as

$$\delta_j = \frac{dE_n}{da_j} = \sum_{k=1}^{K} \frac{dE_n}{da_k}\frac{da_k}{da_j} \qquad (5.19)$$

but if we make use of the definition of δ in order to write $\frac{dE_n}{da_k} = \delta_k$ and

$$a_k = \sum_{k=1}^{K} w_{ji}z_j \xrightarrow{z_j = h(a_j)} \sum_{k=1}^{K} w_{ji}h(a_j)$$

we get to write Equation (5.19) as

$$\delta_j = \dot{h}(a_j)\sum_{k=1}^{K} w_{kj}\delta_k. \qquad (5.20).$$

The Equation (5.20) describes how errors are being backpropaged in the network in order to obtain the gradient information. Note here that the error function can be either the result of the forward propagation of one input vector or the cumulative error of a *batch* of input vectors. The first case is called *sequential* optimization while the latter, where the weights are updated based on the gradient information from the cumulative error of a batch of input vectors, is called *batch-method* optimization. We shall see later how the selection of the batch size affects the model's performance and computation time.

In the existing literature there are plenty of algorithms that, based on the principal ideas of the error backpropagation and the gradient descent, tackle the problem of training ANNs or equivalently minimizing error functions. The most popular, first-order training algorithms or *optimizers* for the time being (Ruder, 2017), are briefly reviewed below:

- **Stochastic gradient descent** (SGD) (and batch or mini-batch methods). This optimizer is one of the first and most successful methods for training ANNs that works with the presented gradient decent and error backpropagation but has serious disadvantages. The learning rate should be properly selected and is the same for all the weights. This fact creates certain issues related to inefficient update of the weights' values and getting trapped on local minima (too small learning rate) or not achieving convergence (too large learning rate). All the following optimizers are either augmented and/or modified versions of the SGD.
- **Nesterov accelerated gradient** (NAG) is one of the optimizers that introduce a momentum term (see next paragraph for the analytical expression of the momentum) and adds it to the gradient information. In methods that use the momentum term an additional coefficient appears that acts similarly to a dumping coefficient and usually takes a value by convention, according to the experience of the community of scientists. Furthermore, NAG estimates the gradient with respect to an approximation of the updated weights, and not in the current weight space position.
- **Adaptive Gradient** (AdaGrad) is an algorithm that adapts the learning rate to its individual adaptive parameter and performs larger updates for infrequent and smaller updates for frequent parameters. This attribute makes AdaGrad very efficient in training ANNs over high-dimensional and sparse data. It achieves to adjust the learning rate of each adaptive parameter by introducing a decaying factor that divides the initial learning rate by a quantity that grows proportionally to the sum of squares of the past gradients.
- **Adadelta** is an extension of AdaGrad that aims to resolve the issue of vanishing learning rate, that appears after a number of training steps with AdaGrad. It defines a window *w* over which the past values of gradient for each parameter are preserved and only their average at the current step is utilized for the update of the learning rate. The exact update rule and other details can be found at the original paper (Zeiler,2012).
- **Adam** (Adaptive momentum estimation) (Kingma & Lei Ba, 2015) is the most recent, popular optimizing algorithm for training ANNs. It expands the idea of adjustable learning rate to the momentum term, as its name reveals. Adam will be presented analytically because is the optimizer used for the training of all the networks in this study. The following paragraph develops the equations that Adam utilizes for the training process.

  Previously the momentum term was mentioned but the analytical expression of an update rule that includes this term is given in Equation (5.21),

  $$u_t = \gamma u_{t-1} + \eta \nabla_\theta E(\theta) \qquad (5.21)$$

  where u is the update vector that is added to the current weight-vector θ, in order to update the weights' values, γ is the momentum coefficient and the right-hand term is the well-known gradient term. However, since Adam belongs in the family of algorithms that assign a weight-specific learning rate, the gradient term is re-defined as

  $$g_{t,i} = \nabla_{\theta_i} E(\theta_{t,i}) \qquad (5.22)$$

where the learning rate η is omitted from the Equation (5.22) because it remains in Equation (5.21). The adjustment of the learning rate takes place in an additional equation, where η is now just the initial value of the learning rate. The subscripts t and i refer to the current time step of the training processes and the i-th adaptive parameter (weight) of the network, respectively.

According to Adam algorithm a set of momentum equations is introduced

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \mathrm{g}_t$$

(5.23)

$$u_t = \beta_2 u_{t-1} + (1 - \beta_2) g_t^2$$

where m_t and u_t are actually estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradient, respectively. The moments are initialized to zero, but this affects the evolution of the training since both terms are now biased towards zero. For this reason, the authors of Adam apply bias correction to the moments with another set of equations

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$$

(5.24)

$$\widehat{u_t} = \frac{u_t}{1-\beta_2^t}$$

and after this step comes the weights' update rule, which has similar form to Adadelta's,

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{u_t}} + \epsilon} \widehat{m_t}. \qquad (5.25)$$

The values of the parameters β_1, β_2 and the infinitesimal ϵ for the implementation of the algorithm are set according to the authors' proposal.

Overall, Adam seems to outperform all the previous optimization algorithms. It is robust and well-suited for a wide range of non-convex optimization problems in the field of machine learning. Hence, it was preferred for the training of the ANN models of this study. Supporting evidence for the above claims can be found in the original paper.