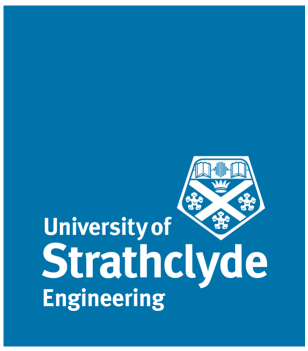

An Integrated Machine Learning Framework for Enhanced Vessel Operational Efficiency

by

Christos Gkerekos

A thesis presented in fulfilment of the requirements
for the degree of Doctor of Philosophy

Department of Naval Architecture, Ocean & Marine Engineering
University of Strathclyde, Glasgow

2020

Declaration of Authenticity and Author's Rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Christos Gkerekos

Date: 24th September 2020

Acknowledgements

> One day, in retrospect, the years of struggle will strike you as the most beautiful.
>
> Sigmund Freud

Firstly, I would like to express my sincere gratitude to Dr Iraklis Lazakis for supervising and funding this study. He patiently instilled in me the virtues of rigorous academic research whilst encouraging me to follow my personal research path. I would like to also thank my second supervisor, Dr Gerasimos Theotokatos, for his valuable support and feedback through the years. Special thanks must be directed to the EU FP7 INCASS project and the Innovate UK ISEMMS project for financially supporting this study.

Besides, this research was enriched significantly through helpful discussions with my colleagues Dr Konstantinos Dikis and Mr Michail Cheliotis. Likewise, I would like to acknowledge the valuable feedback provided by Prof. Peilin Zhou, Prof. Panagiotis Kaklis, and Dr Andrea Coraddu during the annual reviews of this study.

Away from work, special thanks must go to my family, Artemis, Panagiotis, and Sophia for their untiring love, support, and confidence in my every endeavour. Moreover, thanks go to all my friends, and especially to Alexandros Priftis and Christina Petrakopoulou, for their support and for providing a happy distraction to rest my mind outside of my research.

Research Outputs

The following research outputs have been delivered as part of this research work.

Journal publications

1. Christos Gkerekos, and Iraklis Lazakis. A novel, data-driven heuristic framework for vessel weather routing. *Ocean Engineering* 197, 2 2020.

2. Michail Cheliotis, Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. A novel data condition and performance hybrid imputation method for energy efficient operations of marine systems. *Ocean Engineering* 188, 9 2019.

3. Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. Machine learning models for predicting ship main engine Fuel Oil Consumption: a comparative study. *Ocean Engineering* 188, 9 2019.

4. Iraklis Lazakis, Christos Gkerekos, and Gerasimos Theotokatos. Investigating an SVM-driven, one-class approach to estimating ship systems condition. *Ships and Offshore Structures*, 14(5):432–441, 7 2019.

Book chapters

1. Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. Implementation of a self-learning algorithm for main engine condition monitoring. In Carlos Guedes Soares and Ângelo Palos Teixeira, editors, *Maritime Transportation*

and Harvesting of Sea Resources, pages 981–989. CRC Press, 2017. ISBN 978–0815376118

Conference publications

1. Christos Gkerekos, Iraklis Lazakis, and Stylianos Papageorgiou. Leveraging big data for fuel oil consumption modelling. In *Proceedings of COMPIT 2018, the 17th Conference on Computer and IT Applications in the Maritime Industries*, pages 144–152, 5 2018.

2. Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. Exploiting machine learning for ship systems anomaly detection and healthiness forecasting. In *Proceedings of the 2018 RINA Smart Ship Technology Conference*, pages 1–6, 1 2018.

3. Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. Ship machinery condition monitoring using performance data through supervised learning. In *Proceedings of the 2017 RINA Smart Ship Technology Conference*, pages 105–111, 1 2017.

4. Christos Gkerekos, Iraklis Lazakis, and Gerasimos Theotokatos. Ship machinery condition monitoring using vibration data through supervised learning. In Iraklis Lazakis and Gerasimos Theotokatos, editors, *Proceedings of MSO 2016, International Conference on Maritime Safety and Operations*, pages 103–110, 10 2016.

Invited speeches & presentations

1. 2nd ISEMMS project workshop, London (2019). Presentation on ship predictive maintenance and performance monitoring in the digital era.

2. 1st ISEMMS project workshop, London (2017). Presentation on condition monitoring and its applications in a maritime context.

3. LOMAR Shipping, London (2017). Presentation on the use of data-driven Fuel Oil Consumption modelling for vessel performance monitoring.

Contents

List of Figures

List of Tables

Abbreviations

AAKR Auto Associative Kernel Regression

AC Air Cooler

ADLM Automated Data Logging & Monitoring

AE Autoencoder

AIS Automatic Identification System

ANFIS Adaptive Neuro-Fuzzy Inference System

ANN Artificial Neural Network

BBN Bayesian Belief Network

BTA Boosting Tree Algorithm

CAPEX Capital Expenditure

CART Classification And Regression Trees

CBM Condition-Based Maintenance

CFW Cooling Fresh Water

CLT Central Limit Theorem

CM Condition Monitoring

CMEMS Copernicus Marine Environment Monitoring Service

CODLAG COmbined Diesel eLectric And Gas

COM Centre Of Mass

D/Gen Diesel Generator set

DAQ Data Acquisition

DNN Deep Neural Network

DSS Decision Support System

DT Decision Tree

DTR Decision Tree Regressor

E-LLSVM Extended Least Squares Support Vector Machine

EEOI Energy Efficiency Operational Indicator

EG Exhaust Gas

ELM Extreme Learning Machine

EM Expectation Maximisation

ETR Extra Trees Regressor

EWMA Exponential Weighted Moving Average

FMEA Failure Mode and Effect Analysis

FMECA Failure Mode, Effects and Criticality Analysis

FOC Fuel Oil Consumption

GHG Green House Gas

GMM Gaussian Mixture Model

GP Gaussian Process

GTE Gas Turbine Engine

H-SVM Hierarchical Support Vector Machine

HHT Hilbert-Huang Transform

HVAC heating, ventilation, and air conditioning

IACS International Association of Classification Societies

ICE Internal Combustion Engine

IMO International Maritime Organisation

INCASS Inspection Capabilities for Enhanced Ship Safety

IOT Internet of Things

IQR Interquartile Range

ISM International Safety Management

ISO International Organization for Standardization

K-ELM Kernel-based Extreme Learning Machine

KNN K-Nearest Neighbours

KPI Key Performance Indicator

L.O. Lube Oil

LASSO Least Absolute Shrinkage and Selection Operator

LDA Linear Discriminant Analysis

LLSVM Least Squares Support Vector Machine

LLSVM-CIL Least Squares Support Vector Machine for Class Imbalance Learning

LR Linear Regression

LS Least Squares

M/E Main Engine

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MC Markov Chains

MCA Minor Component Analysis

MCC Matthews Correlation Coefficient

MedAE Median Absolute Error

MI Multiple Imputation

MICE Multiple Imputation by Chained Equations

ML Machine Learning

MLR Multiple Linear Regression

MR Multiple Regression

MSE Mean Squared Error

MSLE Mean Squared Logarithmic Error

NaN Not a Number

OC-ELM One-Class Extreme Learning Machine

OC-SVM One Class Support Vector Machine

OCC One Class Classification

OCSVC One-Class Support Vector Classifier

OEM Original Equipment Manufacturer

PCA Principal Component Analysis

R-LLSVM Robust Least Squares Support Vector Machine

RBF Radial Basis Function

RF Random Forest

RFR Random Forest Regressor

Ro-Ro Roll-on/Roll-off vessel

RR Ridge Regression

RUL Remaining Useful Life

SCADA Supervisory Control and Data Acquisition

SFOC Specific Fuel Oil Consumption

SOM Self-Organising Map

SVC Support Vector Classifier

SVDD Support Vector Data Description

SVM Support Vector Machine

SVR Support Vector Reggressor

T/C Turbocharger

TPE Tree-structured Parzen Estimator

Abstract

Inadequate machinery maintenance and inefficient sailing performance comprise two major hindrances to vessel operational sustainability and profitability. To ensure that vessel operation remains competitive while its environmental impact is mitigated, the development of a systematic approach for vessel monitoring and operational enhancement is required. Currently, the maritime industry predominantly operates on a hybridisation of corrective and preventive maintenance, along with monitoring and decision-making based on past experience. More intelligent, data-driven approaches are slowly permeating the industry; these offerings however remain largely rudimentary, retaining considerable assumptions and data requirements for their application. In this respect, this thesis aims to enhance operational efficiency in the maritime industry through the development of an integrated machine learning framework combining efficient and robust machinery anomaly detection, vessel performance degradation monitoring, and routing decision support. This is achieved through a number of key objectives, including: a) the identification of research gaps; b) the extraction of meaningful information for available data sources; c) the monitoring of machinery condition and detection of incipient anomalies; d) the identification of optimal data-driven Fuel Oil Consumption (FOC) modelling architectures; e) the monitoring of vessel performance based on FOC modelling; the facilitation of optimal routing through a suitable Decision Support System (DSS); and f) the demonstration and validation of the above through appropriate case studies. The proposed aim and objectives are accomplished through the combination of a robust pre-processing methodology with a number of data-driven modelling

methods (e.g. One-Class Support Vector Classifiers (OCSVCs), Deep Neural Networks (DNNs)), and a novel modification of Dijkstra's algorithm. A key novelty aspect of this proposed framework is derived by the development and combination of a number of data-driven methodologies for the operational efficiency enhancement of a vessel. Moreover, a novelty of the approach lies upon the minimisation of the inherent assumptions required, streamlining its use in a diverse set of applications. In the same vein, a novel aspect of the proposed framework concerns its flexibility to operate using datasets from different sources, exhibiting different levels of granularity and frequency. This framework is applied to a number of case studies, covering data pre-processing, engine condition monitoring, a FOC modelling comparison, FOC-based performance monitoring, and optimal routing. This helps verify the framework's robustness in a range of realistic scenarios applicable to a variety of vessel types (e.g. reefer, containership, bulk carrier). These case studies, among others, demonstrated the robustness of the anomaly detection methodology when examining different parameters and systems, the accuracy deviation when predicting a vessel's FOC using Automated Data Logging & Monitoring (ADLM) or noon-report data and the optimal models for each case, a successful evaluation of the performance monitoring methodology as a vessel's fouling increases; and the identification of optimal routes as a vessel sails from the Gulf of Guinea to Marseille anchorage.

Keywords: data-driven modelling; fuel oil consumption prediction; condition monitoring; weather routing; anomaly detection

Chapter 1

Introduction

Background information related to this thesis is presented in this chapter. Initially, an introduction to the shipping industry is provided, including its current outlook and potential downturn risks. Following this, a short outline of the importance of monitoring, optimised maintenance and ship optimal routing in the maritime sector is included, including some crucial challenges. Furthermore, the path that the marine industry is following towards digitalisation and data-driven decision-making is discussed, including requirements, opportunities, and challenges. Finally, an outline of the chapters comprising the thesis is included, introducing the reader to the core structure of the thesis.

1.1 Status quo of the maritime industry

Ships are a crucial asset of the global goods transportation system as 85% of merchandise is carried by sea, reaching 11.9 billion tonnes carried during 2018 (Clarkson PLC, 2018). Currently, the 150,000 vessels that comprise the global fleet are valued at US$1.2 trillion (Clarkson PLC, 2018). A break-down of the fleet value per ship type is shown in Figure 1.1. Notably, three-quarters of the global fleet value are attributed to offshore vessels (22.4%), bulk carriers (18.4%), tankers (13.2%), gas carriers (9.1%)

and containerships (10.6%) (Clarkson PLC, 2018).

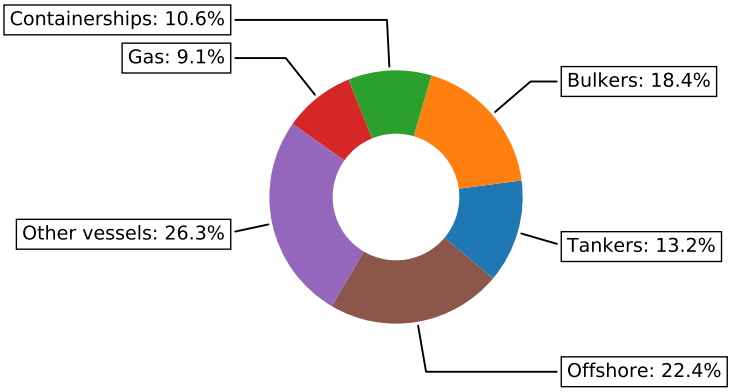

Figure 1.1: Break-down of global fleet 2018 value per vessel type. Adapted from Clarkson PLC (2018).

In spite of that, UNCTAD (2018) reports that in terms of deadweight, about three-quarters of the fleet is represented only by bulk carriers (41.2%) and oil tankers (28.2%). This is visualised in more depth in Figure 1.2.

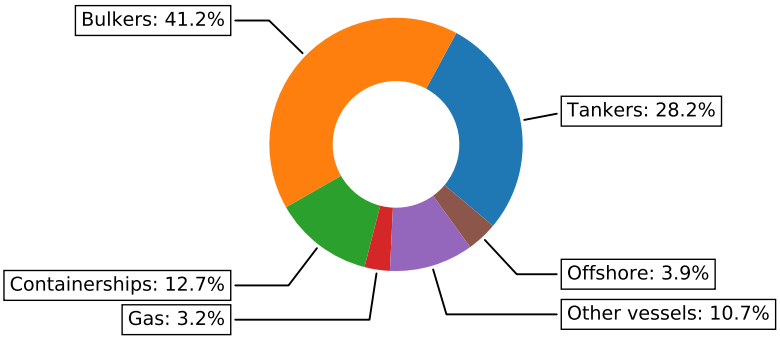

Figure 1.2: Break-down of global fleet 2018 deadweight per vessel type. Adapted from UNCTAD (2018).

Figure 1.3 represents the global fleet distribution in terms of sheer gross tonnage and type. Specifically, 43% of the global fleet size are of medium ($500 \leq$ GT $< 25{,}000$) gross tonnage, 37% are of small (GT < 500) gross tonnage, 13% of large ($25{,}000 \leq$ GT $< 60{,}000$) gross tonnage and 7% are of very large (GT $\geq 60{,}000$) gross tonnage

(Equasis, 2018). Moreover, more than half of the global fleet in terms of number of vessels belongs to the "other" category when inspected from a value or deadweight perspective (Equasis, 2018).

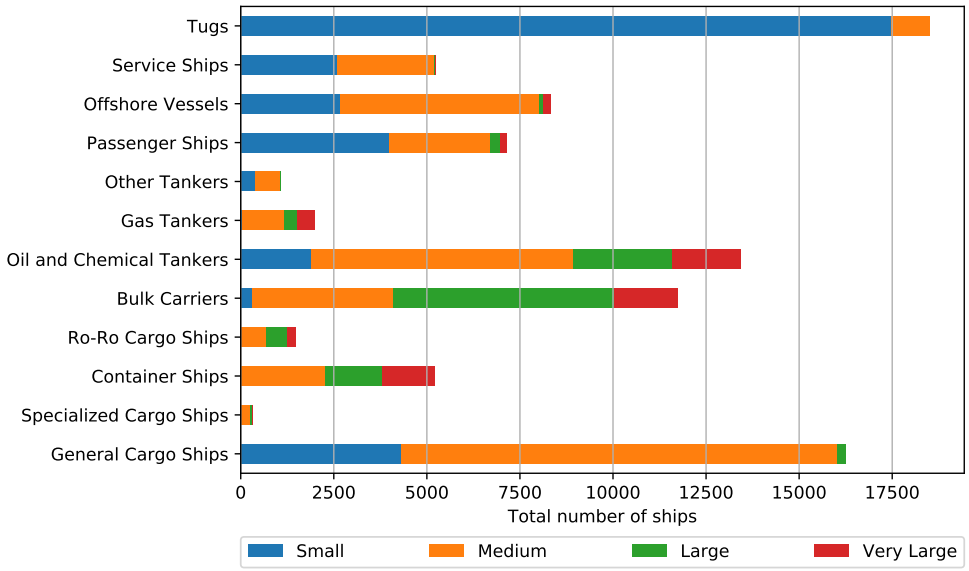

Figure 1.3: Global fleet 2018 total number of ships by type and size. Small corresponds to GT $<$ 500, medium to 500 $\leq$ GT $<$ 25,000, large to 25,000 $\leq$ GT $<$ 60,000 and very large to GT $\geq$ 60,000. Adapted from Equasis (2018).

Over the last five years, shipping growth had been decelerating, mostly due to the Great Recession of the late 2000s and early 2010s that pushed most of the world's developed economies into recession. However, over 2017 an improvement in world fleet expansion was observed, with 42 million gross tonnes being added to the global tonnage, representing a 3.3% growth rate (UNCTAD, 2018). This growth results from a combination of increased newbuilding deliveries and declining demolition activity. At the same time, as seaborne trade volumes increased at a higher pace compared to global tonnage, freight rates increased notably. Over the next five years, shipping outlook projections remain promising, with a relatively stable GDP expansion that is expected to trickle down a similar upswing in merchandise trade volumes (International Monetary Fund, 2019).

However, a consolidation currently observed in shipping companies (UNCTAD, 2018) shows a path where a few large players in the shipping market grow further while observing increased profitability while smaller players struggle to remain afloat. This is compounded by crude oil prices that are forecast to increase over the next decade, as shown in Figure 1.4, putting shipping under further financial strain. Furthermore, IMO enforced a global 0.50% fuel sulphur content cap regulation from 1 January 2020 (MEPC, 2019). Application of the sulphur content cap is expected to have a significant additional impact on fuel costs (Moore Stephens LLP, 2018), exacerbating the increase in bunkering costs. Moreover, other downturn risks appearing on the global horizon such as the adoption of increasingly inward-looking policies and the rise of trade protectionism may also affect seaborne trade growth over the next several years (UNCTAD, 2018).

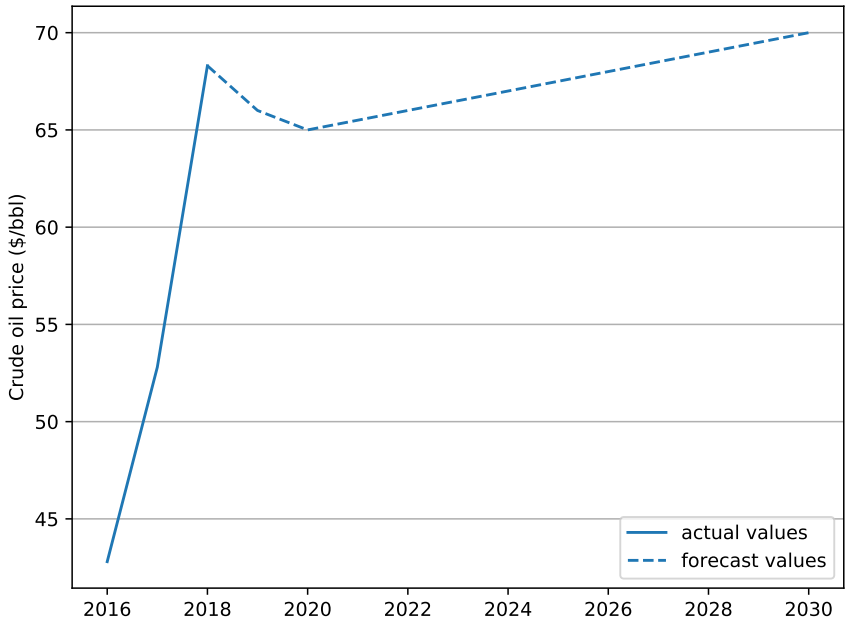

Figure 1.4: Actual and forecast crude oil prices over the 2019–2030 period. Based on World Bank Group (2019) data.

1.2 The impact of monitoring, optimised maintenance and routing

In this context, a new era of shipping is emerging, where technological advancements become the leading force towards extensive condition monitoring, optimised maintenance intervals, spare parts inventory reduction, increased safety, enhanced performance, optimised human (marine and maintenance crew) and financial resources, and reduced environmental impact. This will, in turn, improve the obtainable operational efficiency and permit operators to remain sustainable and profitable even while the industry's profit margins shrink. Two key factors that affect a vessel's operational efficiency are its physical condition and its current operational profile and route, considering external factors such as ambient weather conditions.

The physical condition of a vessel can affect the efficiency of its operations in several ways such as emergency break-downs, risk of increased environmental impact, increased Fuel Oil Consumption (FOC), reduced crew morale and increased fatigue, and relevant safety implications. When the physical condition of a vessel is diminished, maintenance is considered the mitigative action. A number of high-profile marine accidents and near-misses have been attributed to inefficient monitoring and maintenance or complete lack thereof. Such examples include Alexander L. Kielland rig capsize (1980), MTS Oceanos sinking (1991), MV Erika (1999) and MV Prestige (2002) sinkings – both causing unprecedented damage to the European marine environment, MSC Opera crashing into another boat (2019), and the Viking Sky power blackout (2019). At the same time, an increasing number of maintenance-related accidents are attributed to machinery faults. Allianz (2019) note that almost 9000 machinery damage incidents have been observed over the last decade, corresponding to an increase by a third over that period and responsible for US$1 billion worth of insurance claims in five years.

Unplanned maintenance constitutes almost one-fifth of the overall operating costs of a vessel and overall maintenance combined with stores and lubricants contributes approximately one-tenth of the overall running cost of a vessel (Stopford, 2009). Moore

Stephens LLP (2018) report an expected maintenance costs increase by 2–3% year-by-year for the next two years. Additionally, maintenance costs of a vessel increase with its age (Stopford, 2009). As depicted in Figure 1.5, while large and very large vessels tend to be 0-14 years old, small and medium-sized vessels, i.e. the majority of the current fleet, tend to be 15+ years old. Furthermore, as shown in Figure 1.6, the average vessel age has been increasing for the last decade, with over half the global fleet currently being over 15 years old. This denotes a clear forecast of soaring maintenance costs over the next years.

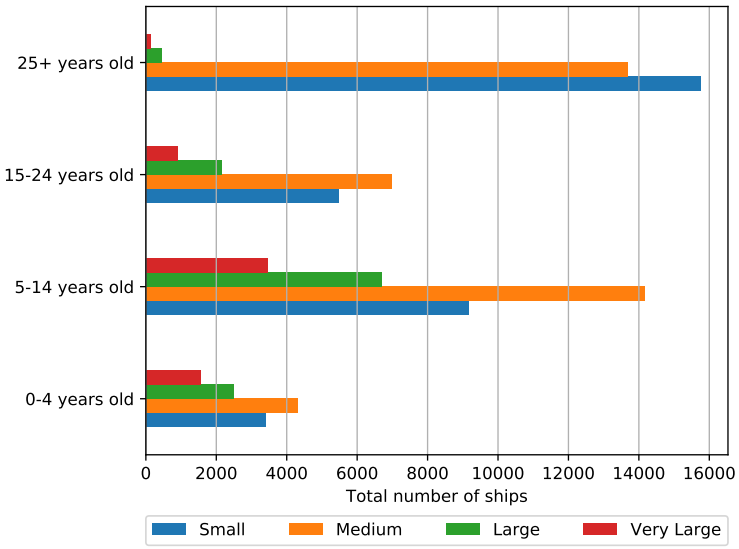

Figure 1.5: Global fleet 2018 age distribution by size. Small corresponds to GT $<$ 500, medium to 500 $\leq$ GT $<$ 25,000, large to 25,000 $\leq$ GT $<$ 60,000 and very large to GT $\geq$ 60,000. Adapted from Equasis (2018).

In sectors such as defence, aviation, manufacturing, automobile, and nuclear power generation, maintenance focus has shifted from reactive and preventive towards predictive. However, in the maritime sector, ship maintenance has been considered an area of excessive expenditure and advanced monitoring methods have not yet been widely applied (Lazakis & Ölçer, 2016; Lazakis et al., 2010). Nevertheless, attempts towards predictive maintenance in shipping have been made in the past years and are rapidly progressing.

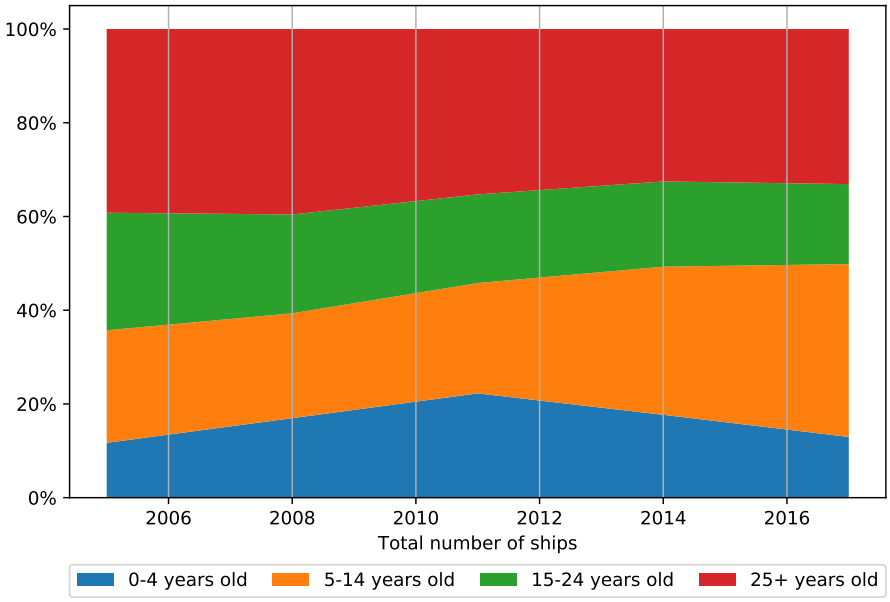

Figure 1.6: Global fleet age distribution by year. Based on Equasis (2006, 2009, 2012, 2015, 2018) data.

Besides the physical condition of a vessel and its relevant sailing performance, sailing efficiency is also an important factor that can affect a vessel's operating costs. Sailing in adverse weather conditions can increase FOC and CO_2 emissions by over 50% (Prpić-Oršić et al., 2016). Furthermore, FOC constitutes approximately two-thirds of a vessel's voyage costs (Stopford, 2009). Therefore, identifying a ship route that avoids adverse weather can yield significant benefits in both financial and environmental terms.

Routing has been traditionally performed based on expert judgement, tapping on relevant experience in combination with any available data, such as weather forecast reports. Due to the effect an optimal route has on a vessel's operating costs and emissions footprint (Zis et al., 2020), it is therefore pursued by a number of maritime industry stakeholders such as ship operators, maritime regulators, and policy-makers. This makes it an active research topic where different modelling approaches are combined with optimisation techniques aiming at different approaches to optimal routing.

1.3 The path towards digitalisation & data-driven decision-making

Over the last few years, companies in transport & logistics have started implementing data-driven technologies (commonly known as Internet of Things (IOT)) in diverse settings, including, *inter alia*, predictive asset maintenance, and route optimisation (Deloitte, 2015). Furthermore, a recent study conducted by Inmarsat (2018) found that three-quarters of shipowners planned to deploy IOT solutions over the next 18 months at an average, per-business, expenditure of US$2.5 million over a three-year period. Three-quarters of the survey respondents expected to realise or had already realised savings due to the implementation of IOT solutions. Notably, over half of the respondents noted that route optimisation constitutes a key reason for the implementation of such solutions.

Digitalisation will be a key driving force for the marine industry over the next years with autonomous ships, drones, and block-chain applications all appearing on the radar (UNCTAD, 2018). However, establishing safety, security and reliability remain pivotal in getting governmental approval – a prerequisite for widespread installation and use. Due to this, the trend towards data-driven decision-making is expected to continue over the next years with new policies coming into force, requiring more stringent Data Acquisition (DAQ) practices. Since the beginning of 2019, vessels of over 5000 GT are required to have a DAQ installed for FOC monitoring as a corollary of the adoption of amendments to MARPOL Annex VI, Chapter 4 (IMO, 2016). International Maritime Organisation (IMO) adoption of an initial strategy in 2018 to reduce greenhouse gas emissions by 50% by 2050 compared to 2008 will have a similar effect as increased operational efficiency will be pivotal in achieving these goals.

Bearing the above in mind, an effective operational efficiency enhancement framework should be able to monitor a number of parameters through various data pipelines, provide an estimation of a vessel's current condition, issue advanced warnings relating to incipient faults and degradation, and quantitatively support decision-making

regarding vessel operations and routing. In this respect, the proposed research framework considers data from sources such as weather forecast reports, IOT sensor data, and noon-report data and employs a hybrid approach to monitor the condition of machinery and the vessel's overall performance to issue warnings regarding incipient faults and degradation and offer decision support for optimal routing.

1.4 Thesis layout

The thesis layout introduces the reader to the flow and structure of each chapter of the thesis. The thesis is structured into 7 chapters summarised below.

Chapter 1: Introduction This chapter sets the scene for the thesis. An introduction to the shipping industry, importance of monitoring, optimised maintenance and routing, and data-driven decision-making in a maritime context are provided. This helps deliver the incentive for the proposed research framework.

Chapter 2: Research Aim & Objectives Chapter 2 includes the research question, main aim and objectives of this thesis. The included objectives provide an overview of the critical research challenges to be addressed in order to attain the main aim of this thesis.

Chapter 3: Critical Literature Review This chapter provides a critical review of research literature concerning maintenance, condition-monitoring, optimal routing, and data-driven decision-making. This review will help uncover research gaps and past trends to define and develop the main research concept of this thesis.

Chapter 4: Methodology & Modelling This chapter describes the considered and developed framework and methodologies of this thesis, focusing on their modelling principles. These methodologies and overall framework stem from the research gaps identified in the previous chapter. All stages of the proposed operational efficiency enhancement framework are considered, including data pre-processing, condition

monitoring, vessel FOC-based performance monitoring, and optimal routing decision support.

Chapter 5: Case Studies' Description Chapter 5 presents the specifications of all case studies included in this thesis. All stages of the proposed framework are validated through case studies, including data pre-processing, detection of incipient machinery faults, vessel sailing performance monitoring, and weather routing.

Chapter 6: Case Studies' Results & Discussion This chapter presents and discusses the results of the case studies presented in Chapter 5. The outcomes of each case study are discussed both individually and as part of the greater framework presented in Chapter 4. The results and their discussion validate the viability and applicability of the proposed research framework.

Chapter 7: Discussion & Conclusions Chapter 7 discusses the overall findings of this thesis along with some concluding remarks. Specifically, key research findings are summarised and the novelty of the thesis is highlighted. Furthermore, the provided conclusion summarises the key learning points of the research performed. Finally, recommendations for future research are provided, taking into consideration the key outcomes obtained and future research and industry trends.

1.5 Chapter summary

In this chapter, background information on the shipping industry was provided, including monitoring, optimised maintenance, and routing. Various requirements, opportunities and challenges were described. Moreover, an outline of the chapters forming this thesis was included to introduce the reader to the flow and structure of the thesis. In the following chapter, the research question, aim, and objectives of this thesis will be presented, demonstrating the ambitions and expectations of the research performed.

Chapter 2

Research Aim & Objectives

This chapter formulates the research question posed and describes the main aim and objectives of this present thesis.

2.1 Research question

The research question of the present thesis is formulated as:

How to develop and implement a realistic strategy to enhance operational efficiency in the maritime industry utilising data-driven methods for machinery condition monitoring and identification of incipient anomalies, and vessel performance degradation monitoring, whilst also providing optimal routing decision support?

2.2 Aim & objectives

The main aim of the thesis is to enhance operational efficiency in the maritime industry through the development of an integrated machine learning framework combining efficient and robust machinery anomaly detection, vessel performance degradation monitoring, and routing decision support.

The objectives through which the main aim of this thesis will be achieved are listed

below:

1. Identify gaps pertinent to the research topic through the critical review of literature pertinent to data-driven monitoring, optimised maintenance and routing.

2. Consider and address the identified gaps through the development of a streamlined methodology that enhances vessel operational efficiency.

3. Extract meaningful information from available data sources through the development of a suitable data pre-processing methodology.

4. Monitor the condition of machinery and detect incipient anomalies through the development of a suitable data-driven methodology, limiting the number of data-related assumptions.

5. Identify optimal data-driven modelling architectures for the prediction of vessel Fuel Oil Consumption (FOC) through a formalised, novel methodology.

6. Monitor the performance degradation of vessels based on FOC modelling through the development of a suitable data-driven methodology.

7. Facilitate optimal routing through the development of a suitable Decision Support methodology.

8. Demonstrate and validate the performance and applicability of the framework and methodologies developed through case studies reflecting realistic scenarios applicable to a variety of vessel types (e.g. reefer, containership, bulk carrier).

2.3 Chapter summary

In this chapter, the research question of the present thesis is formulated and the main aim and objectives are identified and expressed. In the following chapter, a critical literature review is presented to identify current approaches applied to maritime maintenance, monitoring, and optimal routing, future trends and existing gaps.

Chapter 3

Critical Literature Review

This chapter presents the critical literature review conducted as part of framing the research methodology elaborated in Chapter 4 and establishing its inherent novelty. This critical review starts by providing an introduction to the concept of maintenance, followed by an overview of predominant maintenance strategies, highlighting their similarities and differences, and benefits and shortcomings. Following this, condition monitoring applications in the maritime industry are reviewed and critically compared to contemporary advances of other sectors. As most such advances are based on underlying Machine Learning (ML) model architectures, a high-level description the workings of these models is provided for the benefit of the reader. Wherever required, further elaboration on the mathematical foundations and implementation intricacies of these models will be provided in the Methodology chapter. As a vessel's Fuel Oil Consumption (FOC) can provide interesting insights in its overall sailing performance, this metric can be used as a proxy for performance monitoring Key Performance Indicators (KPIs). Due to this, approaches for the modelling of vessels' FOC are also critically reviewed. Finally, weather routing approaches that can utilise the output of the FOC models are critically reviewed. Existing gaps of the current literature are identified and utilised for the introduction and development of the novel methodology suggested in the methodology chapter of this thesis.

3.1 Maintenance & maintenance strategies

3.1.1 A primer on maintenance

Maintenance in general can be defined as (BS EN 13306, 2017):

> [The] combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function.

Four maintenance strategies are, in general, applicable to machinery applications: reactive, preventive, predictive (condition-based), and proactive. Reactive maintenance concerns maintenance that is only performed when a machinery item fails completely. Preventive maintenance concerns maintenance actions that happen periodically, at a predefined frequency. Predictive maintenance takes into consideration the condition of a machinery item and performs maintenance actions accordingly. Finally, proactive maintenance aims to mitigate against the underlying conditions that lead to failures. The evolution of maintenance strategies, as described above, is visually depicted in Figure 3.1.

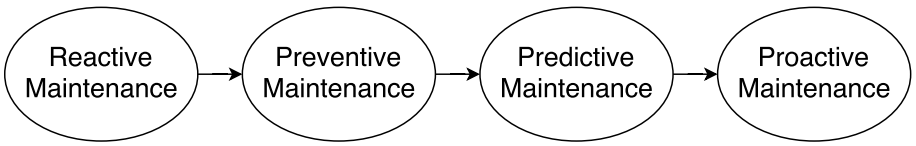

Figure 3.1: Evolution of maintenance strategies.

Stemming from the definition of maintenance presented above, Mehta and Reddy (2015) posit that the commonality of all maintenance strategies is that they aim to minimise (i) component failures; (ii) plant impact from failures; and (iii) overall cost of maintenance.

As multiple decision criteria for the selection of optimal maintenance strategies exist (e.g. cost, reliability, availability and safety (Emovon et al., 2018)) and not all of them can be satisfied concurrently, the way different criteria are prioritised leads to the implementation of a different maintenance strategy. For example, aiming to minimise

the initial cost of maintenance will lead to the implementation of a reactive maintenance policy whereas aiming to minimise the overall cost of maintenance through the life of a system may lead to the implementation of a proactive maintenance policy. In practice, hybrid strategies considering the criticallity of different components and systems are applied. The four maintenance strategies discussed above are presented below in detail, including their inherent trade-offs.

3.1.1.1 Reactive maintenance

Reactive (also known as run-to-failure, breakdown or corrective) maintenance concerns maintenance that is only performed following the complete failure of a component. At that point, no repairing is possible and the component is replaced by a new one (Mohanty, 2017). Nevertheless, Girdhar and Scheffer (2004) and Randall and Antoni (2011) contradict that, stating that repairing is possible, albeit with significantly increased cost as an extensive spare-parts inventory is required. Additionally, costs are only incurred when necessary, as no activity is performed prior to an emergency breakdown.

This method of maintenance provides the longest time between shutdowns, however failures are catastrophic and can possibly affect multiple components and/or machines (Randall & Antoni, 2011). Hence, reactive maintenance is mainly applied to relatively inexpensive and non-critical machines or where redundancies have been implemented so that production is not interrupted. Otherwise, there is a high risk of the production-loss cost eventually exceeding that of maintenance. Reactive maintenance is patently easy to implement, and carries a low initial cost.

Due to the criticality of most systems installed on board a vessel, applications of reactive maintenance are few and far between. However, systems that do not benefit from an advanced maintenance strategy are cabling, piping, lighting, and heating, ventilation, and air conditioning (HVAC) systems along with various redundant components within systems.

Application of reactive maintenance usually leads to limited personnel requirements but high labour costs. This can be justified as immediate and unplanned maintenance is required to follow the failure of a machine. Unplanned maintenance entails overtime work for maintenance personnel. Additionally, this can negatively affect personnel morale as they may become overworked while lacking any predefined work schedule (Girdhar & Scheffer, 2004). However, in some cases this can be partially mitigated by reinvesting some of the money that could have been spent in a more advanced maintenance strategy as an extra payment to the maintenance crew.

Summarising the above comments, the benefits and shortcomings of reactive maintenance are presented in Table 3.1.

Table 3.1: Benefits and shortcomings of reactive maintenance. (Girdhar & Scheffer, 2004; Mohanty, 2017; Randall & Antoni, 2011)

Benefits	Shortcomings
• Longest time between shutdowns as no maintenance is performed in between breakdowns • Costs are only incurred when necessary • Limited personnel requirement due to reduced planning • Easy implementation • Lower initial cost • Additional income for maintenance crew	• More costly in the long term, due to unplanned downtime overtime costs, and income loss • Maintenance can be costly when required • High downtime once maintenance is required • Large spare-parts inventory required to cater for emergency replacements after breakdowns • Reduced machinery item lifespan • Operations unpredictability due to no scheduled maintenance • Possible secondary equipment damage • Reduced personnel morale

3.1.1.2 Preventive maintenance

Preventive maintenance refers to maintenance that happens at a fixed frequency, usually following Original Equipment Manufacturers (OEMs), International Association

of Classification Societies (IACS), or International Safety Management (ISM) code recommendations. Guiding principle of reactive maintenance is that if a machinery item is "well" maintained, no unplanned downtime is to be expected.

Compared to reactive maintenance, preventive maintenance offers significant increase in machinery item lifespan. This is due to the fact that the probability of catastrophic failures is diminished. Additionally, preventive maintenance is more cost-effective as the amount of components or machinery items that need complete replacement is reduced. Moreover, as a considerable tranche of maintenance is performed as a precaution and before the perception of any defects, unplanned downtime is reduced.

Preventive maintenance generally aims to provide such maintenance intervals that only 1-2% of machinery experience failures between maintenance intervals (Randall & Antoni, 2011). Thus, the vast majority of machinery items would be able to continue working without maintenance for two or three maintenance intervals (Neale & Associates, 1979).

However, this has been shown to cause reduced morale in maintenance workers due to time-consuming task scheduling and execution (Randall & Antoni, 2011). Additionally, this introduces increased "infant mortality" in machines, due to faults that would otherwise have been avoided (Randall & Antoni, 2011). Infant mortality concerns both failures caused by faulty replacements and by general tampering during maintenance activities. Besides, excessive maintenance causes significant, albeit planned, downtime. Furthermore, due to the extent of the planned maintenance activities, significant capital is invested in spare parts inventory leading to an additional cost of money to be considered. On top of that, unexpected failures may still occur as maintenance happens at a fixed frequency, without taking into consideration the actual machinery item condition. The model usually used for component deterioration (colloquially known as the "bathtub curve" (Nowlan & Heap, 1978)) is presented in Figure 3.2. This figure demonstrates that the component starts with a relatively high probability of failure due to the effect of infant mortality. As time progresses, this probability decreases

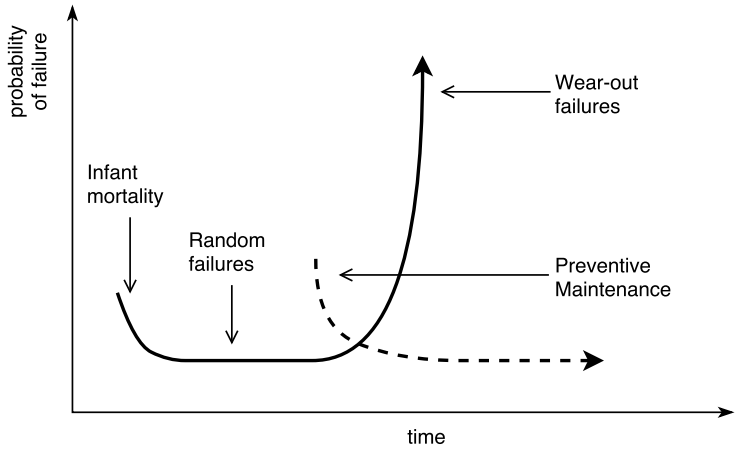

Figure 3.2: Visualisation of component deterioration and timing of associated application of preventive maintenance. Adapted from (Nowlan & Heap, 1978).

and then plateaus out as the risk of infant mortality diminishes while the component is well within its expected life range. At the final stage, as the component nears the end of its expected life range, the probability of failure increases sharply. Application of preventive maintenance starts near the expected end of the plateau. There, the probability of failure incipiently increases due to the reappearance of infant mortality but then decreases back to the levels of the previous plateau.

In the shipping industry, preventive maintenance currently constitutes the maintenance status quo, mainly due to its lower installation cost compared to predictive maintenance. Based on this strategy, critical components are maintainted and replaced based on a maintenance schedule. Examples of machinery systems where preventive maintenance is applied are the majority of main engine components, inlcuding cylinder liners, piston rings, and valves.

Summarising the above comments, the benefits and shortcomings of preventive maintenance are presented in Table 3.2.

3.1.1.3 Predictive maintenance

Predictive maintenance (also known as Condition-Based Maintenance (CBM)) provides a more intelligent method of maintenance planning. There, present and past condition

Table 3.2: Benefits and shortcomings of preventive maintenance. (Neale & Associates, 1979; Randall & Antoni, 2011)

Benefits	Shortcomings
• Machinery item life span increased • More cost-efficient than reactive maintenance • Unplanned downtime reduced	• Time-consuming task scheduling • Significant planned downtime due to excessive maintenance • Unexpected failures can still occur • machinery item condition not taken into consideration, leading to excessive maintenance • Incidental damage may arise from unneeded maintenance • Excessive tied-up capital in spare parts inventory

of each component is taken into consideration in order to offer bespoke maintenance scheduling for each system and component.

Predictive maintenance requires a higher expenditure at installation but over an extended period of time, becomes more economical than preventive or reactive maintenance. Especially in industries where machinery items are expected to run for long periods without any shutdowns, it has been shown that predictive maintenance can reduce relevant costs by up to 65% (Neale & Associates, 1979). Furthermore, in terms of downtime, planned downtime is minimised to the bare necessary minimum and unplanned is almost diminished. This optimised maintenance scheduling permits the maximisation of machinery item lifespan.

In order to estimate past and present condition of components and machines, different types of measurements are acquired and processed. Girdhar and Scheffer (2004) and Mohanty (2017) note that, depending on application and data available, main types of input data comprise:

Vibration measurements constitute the most reliable technique to identify mechanical defects in rotating machinery. However, specialised equipment is required

for Data Acquisition (DAQ) so for most machinery items this is not applied as a fixed monitoring system but through on-the-spot measurements and analysis of data from a specialised crew utilising portable equipment.

Acoustic measurements are primarily used for structural monitoring but some applications for the monitoring of machinery have emerged in recent years.

Oil/debris analysis focuses on analysing the debris detecting in the oil used for the lubrication of machinery items. This can provide valuable information regarding the condition of bearings and gears.

Infrared thermography is used to analyse active electrical and mechanical systems. The identified heat signature can be used to detect defects in systems such as generators, boilers, and shaft and coupling misalignments.

Motor current-signature analysis is used to detect faults in induction motors the analysis of the current-signature produced. Recent applications have extended this analysis and can now also be used in systems such as pumps and gearboxes by analysing the quality of the current drawn by the electric motor.

Performance monitoring takes advantage of the link between the efficiency of machinery systems and problems in their operation.

Visual inspection constitutes the simplest method of monitoring a piece of machinery but benefits from a very low cost of implementation, and benefits from the knowledge that may be already acquired by the maintenance crew.

In the shipping industry, predictive maintenance is mostly provided as a service by Main Engine (M/E) and bearings OEMs. The use of such services remains infrequent and carries a high cost.

Guillén et al. (2016) have produced a list of available International Organization for Standardization (ISO)-standardised Condition Monitoring (CM) techniques. These approaches are shown in Table 3.3.

Table 3.3: ISO CM techniques (Guillén et al., 2016).

CM technique	ISO reference
Vibration	13373-1:2002; 13373-2:2016; 16587:2017; 18436-2:2014
Thermography	18434-2:2019; 18436-7:2014
Acoustic emission & ultrasound	22906:2007; 29821-1:2011; 18436-6:2014
Tribology and lubricants	18436-5:2012; 14830-1 (under development)

Still, predictive maintenance requires a high initial spending. Also, while during a machine's lifespan predictive maintenance proves to be more economical, results take years to show. Furthermore, skilled personnel are required to interpret the results obtained through condition monitoring in order to provide an optimised maintenance schedule.

Summarising the above comments, the benefits and shortcomings of predictive maintenance are presented in Table 3.4.

Table 3.4: Benefits and shortcomings of predictive maintenance. (Girdhar & Scheffer, 2004; Mohanty, 2017; Neale & Associates, 1979)

Benefits	**Shortcomings**
• Reduced maintenance cost over machinery item lifespan • Planned downtime only when required • Unplanned downtime almost diminished • Machinery item lifespan optimised • Increased availability and reliability	• High installation cost due to need for specialised equipment • Skilled personnel required to interpret condition-monitoring results • Results take time to show

3.1.1.4 Proactive maintenance

Proactive maintenance concerns the identification of any underlying root-causes of a fault using well-established engineering knowledge and the provision of a maintenance solution that addresses these latent causes so that the fault is not repeated (Fitch,

1992). In that sense, proactive maintenance can be seen as an extension of predictive maintenance. Due to this, large-scale commercial applications of proactive maintenance are not found in the shipping industry.

Proactive maintenance provides the benefit of further extending the machinery item lifespan by controlling any underlying causes of failure. This reduces maintenance cost and downtime as faults are being avoided instead of just predicted and mitigated. At the same time, proactive maintenance dictates a high preliminary cost as for each detected fault, its root causes have to be identified and analysed (Fedele, 2011). Accordingly, skilled personnel are required to perform the above tasks. Furthermore, results take time to show and are difficult to quantify as faults are being avoided.

Summarising the above comments, the benefits and shortcomings of proactive maintenance are presented in Table 3.5.

Table 3.5: Benefits and shortcomings of proactive maintenance. (Fedele, 2011; Fitch, 1992)

Benefits	Shortcomings
• Maintenance cost reduction • Downtime minimisation • Increased availability and reliability • Enhanced income generation	• Requirement for skilled personnel • High initial cost • Results take time to show and cannot easily be quantified • Requirement and frequent lack of necessary data

3.2 Condition monitoring applications

Following the overview of maintenance strategies discussed above, it is evident that for most critical applications, aspects of predictive and proactive maintenance need to be implemented. An intrinsic requirement for the implementation of a condition-based maintenance strategy, is the ability to accurately monitor the condition of an asset.

In sectors such as defence, aviation, manufacturing, automobile, and nuclear power generation, this maintenance shift has been observed over the past years, and condition monitoring applications have evolved accordingly. However, in the maritime sector, ship maintenance often remains considered an area of needless expenditure and advanced monitoring methods have not yet been widely applied (Lazakis & Ölçer, 2016; Lazakis et al., 2010). Nevertheless, attempts towards predictive maintenance in shipping have been made in the past years and are rapidly progressing.

For the benefit of the reader, prior to the critical review of condition monitoring applications, an overview of the most prominent modelling algorithms with applications in condition monitoring are presented in Appendix A.

Some key applications of condition monitoring are reviewed below. Firstly, applications within the maritime industry are critically reviewed to identify the current state of the art within the inherent complexities that are present in this sector. Following this, other comparable sectors are reviewed, in order to identify and understand future trends and determine relevant research gaps and directions for research.

3.2.1 Maritime industry

A significant number of condition monitoring applications within the maritime industry focus on the main engine as it constitutes a critical component that is required for the vessel to be safe, available and profitable, and where in most cases no redundancies exist. Many such applications utilise first-principle, physics-based models while in recent years data-driven applications are starting to become more commonplace.

For example, a methodology where vibration data are combined with performance data (cylinders' pressure) for the condition monitoring of a main engine has been suggested (Chandroth, 2004). Accordingly, engine thermodynamic models have been developed and used to perform condition monitoring. Specifically, Watzenig et al. (2009) developed thermodynamic models that can be used to detect two common failure models, i.e., increased blow-by, and compression ratio failures. Lamaris and Hountalas (2010)

developed a multi-zone thermodynamic model that can be applied to both two-stroke and four-stroke engines. This offers the ability to operate accurately at different operating conditions without requiring fine-tuning and can used for the diagnosis of several power plant faults. Moreover, Hountalas (2000) developed a diesel engine performance model that can account for both normal and faulty conditions. The key shortcoming of these methods is their inherent requirement for the development of a complex physics-based model that still is only pertinent to a single engine model. Moreover, such models cannot easily incorporate the effects of maintenance and system ageing.

Dikis and Lazakis (2016) presented the framework of the Inspection Capabilities for Enhanced Ship Safety (INCASS) project that developed tools to enhance machinery monitoring by combining real time information with machinery risk analysis tools. In this scope, a machinery risk analysis tool that performs condition monitoring and maintenance decision support through the combination of Markov Chains (MC) and Bayesian Belief Networks (BBNs) was developed. Dikis et al. (2017) further elaborated on this tool by considering and assessing components' risk of failure and reliability degradation by utilising raw input data. The end output of this study was a maintenance Decision Support System (DSS) system that was tested in case studies based on data obtained through a number of vessels (tanker, bulk carrier, and containership). While interesting, the approach proposed in this work is hard to implement as detailed failure rates of different components and subcomponents are required.

Coraddu et al. (2016) and Cipollini et al. (2018a) suggested a regression method for the estimation of component degradation in a marine COmbined Diesel eLectric And Gas (CODLAG) propulsion plant type in the existence of degradation information in the model training dataset. Support Vector Machine (SVM) and Least Squares (LS) models were applied, with both yielding comparable results. In these studies simulated data were used, therefore decay state information was also explicitly available. Cipollini et al. (2018b) then utilised the same dataset to compare supervised and unsupervised algorithms for fault detection. The authors noted that One Class Support Vector Machine (OC-SVM) and a one-class version of the K-Nearest Neighbours (KNN) algorithm

yielded comparable results. Raptodimos and Lazakis (2018) investigated the application of Self-Organising Maps (SOMs), an unsupervised neural network architecture, for monitoring the condition of a two stroke marine diesel engine by identifying clusters containing data representing abnormal engine operating conditions. The authors noted that the SOM architecture delivered promising results in monitoring the condition of a M/E and that this work could be extended by the introduction of diagnostic elements within the proposed framework. Besides, a self-learning (fuzzy Artificial Neural Network (ANN)) algorithm for fault diagnosis in the combustion system of a marine diesel engine has been developed (Li et al., 2010). All the approaches reviewed in this paragraph can effectively identify faults in marine engines through the implementation of data-driven methods. A key drawback however is that, to train the proposed models, a dataset that incorporates multiple recorded instances of each identifiable fault of the examined system are required. Effectively, this means that a significant amount of time will have to pass between sensor installation and having an operational CM system based on these approaches. Moreover, the lack of consensus regarding the optimally performing modelling approach should be noted.

Perera (2016) combined hard measurement thresholds with a Gaussian Mixture Model (GMM) with an Expectation Maximisation (EM) algorithm to identify sensor faults in marine engine data. Data points are then examined under Principal Component Analysis (PCA) to identify sensor faults even in complex situations. The author noted the proposed approach can by utilised even in cases where multiple sensors have failed concurrently. Brandsæter et al. (2019) proposed a Auto Associative Kernel Regression (AAKR)-based signal reconstruction followed by residual analysis for the anomaly detection of a marine diesel engine. The authors noted that this approach requires a reduced computational time and that it can be successfully be applied in the presence of imbalanced datasets.

In this respect, data-driven approaches that can perform engine condition monitoring while minimising the amount of data requirement assumptions should be further investigated.

3.2.2 Other industries

Jardine et al. (2006) have provided a review of diagnostic and prognostic implementations of condition-based maintenance across a number of industries, noting the emerging trend of using multiple sensors as input to CM systems. Since then, as CM applications started expanding and becoming the norm in a number of sectors, review papers started focusing on specific sectors and applications. Along these line, Touret et al. (2018) have provided an overview of approaches utilising temperature measurements to detect gearbox failures. The authors note the increased cost of vibrational analysis and that thermocouple and oil temperature measurements can often be an accurate descriptor of a system's condition. Stetco et al. (2019) have accordingly provided a thorough review of machine learning methods for condition monitoring of wind turbines. They noted that two-thirds of reviewed models utilise classification techniques while the rest utilise regression techniques. Furthermore, the authors noted that even within the field of wind turbine CM there is no single model that outperforms all others in all datasets and tasks.

Due to the this observation and due to the intrinsic modelling differences of performing condition monitoring through classification and regression, these applications are categorised first by the type of the model's output variable (continuous vs. discrete) and then based on the sector that each application applied to. Following the review of regression-based and then classification-based models, one-class classification-based and unsupervised models are reviewed in the following sections.

3.2.2.1 Regression-based models

Regression-based models aim to predict the normal behaviour of different components and subcomponents, assumed to be in a healthy state, and compare those predictions with their actual behaviour to use the obtained similarity index as a proxy for their condition.

Wind turbines Faulstich et al. (2011), focusing on the condition monitoring of wind turbines, have noted that data for normal behaviour models should be obtained while components remain in the flat area ("random failure zone") of the "bathtub curve" component deterioration model depicted in Figure 3.2. Different input variables have been considered for regression-based models, with Schlechtingen et al. (2013) noting that in the case of wind turbine power generation, the use of wind direction and ambient temperature apart from wind speed yielded more accurate results. Accordingly, regarding the interpretation of results, Kriegel et al. (2010) remarked that while sometimes assuming a normal distribution and identifying points more than two standard deviations away from the mean as outliers is accurate, this cannot be generalised and often other approaches need to be investigated. Laouti et al. (2011) applied Support Vector Reggressors (SVRs) with an Radial Basis Function (RBF) kernel to detect different faults in wind turbine components. The authors noted that this approached permitted the detection of converter torque faults within two sample periods but that faults in the actuators of the pitch systems could not be detected. Orozco et al. (2018) utilised big-data distributed processing frameworks to identify wind turbine faults. Due to the large amount of data processed, an automated labelling system was introduced, where residuals above the 99th percentile were flagged and data points that were followed by a wind turbine shutdown were labelled as abnormal. It was found that temperature and power data suffice to identify potential failures. Moreover, in regards to the chosen metrics, the Linear Regression (LR) and polynomial regression models yielded the best performance.

L. Wang et al. (2017) built Deep Neural Network (DNN) models for regression purposes, aiming to predict lubricant pressure based on Supervisory Control and Data Acquisition (SCADA) data, concluding that in this case DNN models performed better than models such as Least Absolute Shrinkage and Selection Operator (LASSO) models, Ridge Regressions (RRs), KNN, SVMs, and shallow ANNs.

Bearings Soualhi et al. (2015) combined the Hilbert-Huang Transform (HHT), i.e. a signal decomposition method, with an SVM model, and an SVR model to obtain

a health estimate, classify the degradation, and estimate the Remaining Useful Life (RUL) of bearings. The authors' results showed increased performance compared to alternative methods.

3.2.2.2 Classification-based models

Classification-based models receive a number of variables as input and assign them one of a number of predefined categories. The number and significance of these categories depends on the labelling available during the training phase. For condition monitoring purposes, categories will either reflect the degradation of a component or system (e.g. not degraded, partially degraded, or fully degraded), the RUL of a component or system (e.g. healthy, 6 months before failure, or 2 weeks before failure), or faults at specific components (e.g. healthy, decreased Lube Oil (L.O.) pressure, or increased Exhaust Gas (EG) temperature).

Wind turbines Leahy et al. (2018) applied all three categorisation scenarios for the fault detection of a wind turbine generator utilising SVM classifiers. Namely, they performed classification to predict system binary condition (fault or no-fault), fault diagnosis, and fault prognosis. While their recall was high, precision was exceptionally low. Santos et al. (2015) applied classification techniques to identify different wind turbine rotor blade faults (e.g. imbalance and misalignment). They found that SVMs with a linear kernel performed best, surpassing the performance of SVMs with a non-linear kernel and ANNs. At the same time, training time was reduced as the amount of hyperparameters to be optimised is also reduced.

Kusiak and Li (2011) followed a similar approach, applying ensemble ANNs for fault identification at different granularity levels, noting that different algorithms performed better at each level. Specifically, ANN ensembles performed best at a high level, whereas Random Forests (RFs) were the ones performing better at an intermediate level, followed by Boosting Tree Algorithms (BTAs) at at the most granular level.

Heat exchangers Casteleiro-Roca et al. (2016) compared Linear Discriminant Analysis (LDA), Decision Trees (DTs), and ANNs for the fault detection of a heat exchanger, noting that a ANN with one hidden layer performed best.

Aircraft engines Q. H. Xu and Shi (2006) applied Hierarchical Support Vector Machine (H-SVM) models for multi-class fault diagnosis in aircraft engines. The authors highlighted that H-SVM models are faster in both training and classification compared to other SVM approaches. Xi et al. (2019) applied Least Squares Support Vector Machines (LLSVMs) models to detect faults in aircraft engines, noting their ineffectiveness in imbalanced datasets. For this reason, they proposed a new implementation of LLSVM called Least Squares Support Vector Machine for Class Imbalance Learning (LLSVM-CIL) that utilises separate C parameters for each class, noting the method's effectiveness in imbalanced datasets. Accordingly, Y.-P. Zhao et al. (2019) evaluated an Robust Least Squares Support Vector Machine (R-LLSVM) approach to aircraft engine fault diagnosis noting its computational complexity. Alternatively, they proposed an improved version of R-LLSVM called Extended Least Squares Support Vector Machine (E-LLSVM) with a lower computational complexity. The authors successfully applied the proposed E-LLSVM for fault detection purposes in aircraft engines.

Y. P. Zhao et al. (2017) applied Extreme Learning Machines (ELMs) for fault diagnosis in aircraft engines. Y. P. Zhao et al. (2019) also applied ELMs for the same problem and suggested an improved version version of the algorithm based on soft margins, noting that the approach seemed promising for the fault detection of aircraft engines. You et al. (2016) applied Kernel-based Extreme Learning Machine (K-ELM) on the problem of aircraft engine fault pattern recognition. Authors noted that K-ELM presents better generalisation capabilities compared to ELM. Following that, they introduced a novel recursive reduced (RR) K-ELM that performed better in engine fault pattern recognition. F. Lu et al. (2017) suggest a similar modification on top of K-ELM algorithm called Dual Reduced (DR) K-ELM algorithm. The authors noted this approach also yielded better results compared to original K-ELM.

Gas Turbine Engines (GTEs) **Hanachi2018** developed an Adaptive Neuro-Fuzzy Inference System (ANFIS)-based system for the fault detection and degradation estimation of a GTE. The authors note that this approach accurately estimated degradation conditions, noting however that accuracy would be increased if data points corresponding to a degraded condition were available during training. ANFIS combines the benefits of ANNs with fuzzy logic, in a unified framework and is often applied to CM tasks (J. Jang, 1991; J. S. R. Jang, 1993). Hanachi et al. (2019) built on top of the aforementioned previous work, combining the ANFIS model with a model-based fault prediction module. The authors remarked that this approach exhibited a ten-fold increase in achieved accuracy when detecting GTE faults.

Sina Tayarani-Bathaie and Khorasani (2015) proposed an ensemble of dynamic ANN models to learn GTE dynamics. Then, the authors used the residuals of these networks as an input to a ANN for GTE fault detection. Simulations performed validated that the proposed approach represents a promising tool. Amozegar and Khorasani (2016) trained ANN and SVM architectures for fault detection and isolation in GTEs. These architectures were combined with ensemble-based techniques, with the authors noting a tangible accuracy increase as a product of this. The models were tested against a simulated GTE dataset and obtained good results.

Automotive Theissler (2017) used a ensemble of two-class classifiers, namely GMM classifiers, Naive Bayes, RFs, and Support Vector Classifiers (SVCs), combined with several one-class approaches to identify known and unknown faults in automotive systems. The ensemble approach performed best, followed by one-class approaches. Wong et al. (2016) proposed an alternative probabilistic ensemble approach, based on Sparse Bayesian ELMs. Authors noted that the results they obtained were improved over the single-classifier baseline. Moreover, the ensemble system was able to detect both single and simultaneous faults while the training set only contained single faults.

3.2.2.3 One-class classification-based models

One-class classification-based models operate based on the same principle as classification models with the caveat that only a minimum amount of data points corresponding to the faulty class are required. Generally, the model draws a boundary that encompasses all points used at training and then evaluates any new point against that boundary. If the new points lie within the boundary they are assigned the same class as the training set, whereas if the new points lie outwith the boundary, they are labelled as anomalous.

Internal Combustion Engines (ICEs) Jung (2019) combined an RF classifier for fault classification with a set of One-Class Support Vector Classifiers (OCSVCs) that can individually identify every predefined fault mode. Moreover, this framework provides a robust method of identifying unknown faults as the OCSVC approach yields significantly better results when previously unseen faults are introduced to the test set, compared to RF classifiers.

Automotive Theissler (2017) used a combination of one-class approaches, namely extreme value, Mahalanobis distance, OCSVC, and Support Vector Data Description (SVDD) as as an ensemble that also took two-class classifiers as input to identify known and unknown faults in automotive systems. The ensemble approach performed best, followed by the OCSVC and SVDD approaches. Two-class classifiers showed significantly decreased robustness in fault detection.

Smart cities Dai et al. (2019) applied multi-layer ANN-based One-Class Extreme Learning Machines (OC-ELMs) to an urban acoustic classification dataset, noting that their proposed algorithm performed better than other representative one-class classifiers such as OCSVCs, Autoencoders (AEs), and Naive Parzen density estimators.

3.2.2.4 Unsupervised learning models

Unsupervised models do not require labelled data for training purposes. In terms of CM and its relevant applications, unsupervised learning is often applied in the form of cluster analysis, where known points are placed in clusters based on commonalities and the commonalities of new points against the pre-derived clusters is evaluated.

Stetco et al. (2019) have noted that even in the, more technologically advanced, renewables sector, unsupervised approaches for condition monitoring have not been explored in depth. Zhang et al. (2019) suggest a combination of subtractive clustering, an one-pass algorithm that estimates the number of clusters and their centres in a dataset, with KNN to identify and classify measurements of unsteady operating conditions. Results showed that this approach performed better in identifying faults in water-source heat-pump systems than traditional methods, such as PCA. Kwan et al. (2003) applied Minor Component Analysis (MCA) to detect and identify faults in aircraft hydraulic pumps. The authors based their fault detection approach on the fact that principal and minor components of sensor measurements will be altered due to the introduction of faults. Moreover the fault size may be estimated, providing a useful input for RUL estimation and fault isolation approaches.

3.2.3 Comparison of data-driven CM approaches

Evaluating the above review of condition monitoring applications, it is clear that CM in other sectors remains more advanced than in marine industry. This observation leads to two key takeaways: a) there is no need to invent new technologies as long as the maritime industry remains a laggard compared to the current state-of-art; and b) the first step to bringing the maritime industry up to speed with the current state-of-art of other industries, approaches that align with the peculiarities of this sector need to be identified.

As a corollary to the above, before the mass deployment of predictive or proactive maintenance strategies in a turn-key fashion, a robust predictive framework is required to be developed. In this respect, and contrary to most other maritime condition mon-

itoring studies reviewed, where either specific dataset requirements have to be satisfied or a first-principles engine model is required, model training should only assume the existence of readily available data points.

Another approach that can achieve this same goal and requirements, is the implementation of a performance monitoring system at a system-level. In the case of the maritime industry, this can be achieved by monitoring the FOC of a vessel while keeping track of any parameters that can affect this parameter (e.g. speed, load condition, weather conditions). The existence of such a model can provide a valuable baseline against which future voyages of the ship can be compared to and its performance be evaluated.

3.3 Data-driven prediction of a vessel's FOC

Bialystocki and Konovessis (2016) performed a statistical analysis of noon-reports of a Roll-on/Roll-off vessel (Ro-Ro) in order to identify the influence of factors such as ship's draft, displacement, weather velocity and direction, and hull and propeller roughness. Once several corrections suggested are applied to the obtained data along with relevant filtering, curves for each frequently-observed sea state are fitted. This provides a simple algorithm that approximates FOC. R. Lu et al. (2015) developed a semi-empirical method for the prediction of operational performance of ships. This method is based on modelling still water and added resistance components. Through that, the ship's operational performance is modelled, taking into consideration the weather and relevant sea state. This model is then utilised to optimise the ship's voyage route.

Beşikçi et al. (2016) suggested the use of ANNs for the prediction of ship fuel oil consumption at various operational conditions. Additionally, a DSS is elaborated for real-time, energy efficient operations. The suggested methodology is compared against Multiple Regression (MR) analysis, displaying superior results. Petersen et al. (2012) evaluated ferry main engine fuel oil consumption modelling approaches, also based on ANNs. The output of the derived models were used for trim optimisation purposes. Meng et al. (2016) suggest a data pre-processing methodology based on outlier-score-

based data. Following that, two regression models were developed in order to link available data with the vessel's fuel oil consumption. The first model connects the ship's fuel oil consumption with its speed and displacement. The second model builds on the first, utilising the information provided by the first while also including weather conditions. They validated the work performed utilising noon-report data from two 13,000-TEU and two 5000-TEU containerships. Simonsen et al. (2018) proposed a method of utilising Automatic Identification System (AIS) data to estimate the fuel oil consumption of cruise ships sailing Norwegian waters. The authors note that the outcome of this method can be used to also estimate Green House Gas (GHG) emissions. Lundh et al. (2016) proposed a method to estimate the fuel oil consumption of vessels equipped with diesel electric propulsion systems. This is used to optimise the use of individual generators in a multi-generator set-up, offering fuel savings of up to 6% when applied to a large cruise ship. Moreno-Gutiérrez et al. (2015) provide a comparative analysis of first-principle approaches to estimating the energy consumption of vessels. Mao et al. (2016) compared linear regression, first-order autoregressive, and a mixed effect models for the speed prediction of a container ship. Accordingly, Yao et al. (2012) investigated the correlation between fuel oil consumption and the ship speed of containerships of different sizes. Reviewing the above publications, it is clear that contradicting conclusions have been reached, with different authors noting that their proposed method performed better than others. Due to this, a thorough review of the potential methods and evaluation their respective results should be conducted.

Cichowicz et al. (2015) provided a methodology for first-principles, time-domain modelling of main and auxiliary engines for assessment of life-cycle ship performance and energy efficiency. Speed and draft are taken into consideration, along with hull fouling and deterioration of engine performance. Sea state is included implicitly by considering an additional M/E load (sea margin). The methodology was demonstrated using data from 3700-TEU containership. Coraddu et al. (2017) performed a comparison of white, grey, and black box models for the estimation of fuel oil consumption of a Handymax chemical/product tanker, concluding that grey-box models can effectively forecast FOC

when only limited historical data are available.

Trodden et al. (2015) focused on data pre-processing and suggest a methodology, ancillary to the ones elaborated above, for splitting available ship data into steady-state chunks that can then be used for fuel efficiency monitoring. The authors noted that steady-state data can be used to identify system degradation over time whilst evaluating both steady-state and transient data are useful in identifying degradation due to the way the vessel is operated. Perera and Mo (2018) suggested another ancillary methodology for the compression of ship performance and navigation data. This is implemented through an AE system, compressing data before transmission and then expanding them upon receipt. Such an implementation is extremely beneficial as the amount of data that can be transferred given any bandwidth and cost constraints is increased, potentially leading to more accurate models. Tsitsilonis and Theotokatos (2018) developed a systematic methodology for energy management of ship prime movers. A statistical analysis is combined with energy and exergy analyses to identify key areas were energy savings can be obtained. This methodology was applied in both Automated Data Logging & Monitoring (ADLM) and noon-report data. S. Wang et al. (2018) proposed a LASSO regression model for the estimation of a vessel's fuel oil consumption. This model was shown to have optimal performance when compared to ANN, SVR, and Gaussian Processs (GPs) models in a case study utilising low-frequency data obtained from a fleet of containerships.

From the above, it can be deduced that modelling of vessels' FOC is an active research field with multiple different approaches being realised concurrently. However, up to the present, most studies utilise different datasets, with different acquisition and modelling particularities hindering any attempts at a comparison. Moreover, the proposed approaches often require significant data filtering before the application of the relevant models, suggesting models that can only operate in a restricted window of sailing conditions and operating profiles.

Having built a model that can accurately model a vessel's FOC over varying sailing and

operating conditions, the usage of this model can reversed to identify the most favourable sailing conditions, i.e. optimal weather routing. Besides, the vast majority of the reviewed papers investigate FOC prediction models on their own, without considering the applications for which a FOC prediction model would be useful. In this respect, applications based on FOC prediction models should be further investigated.

3.4 Weather routing

Walther et al. (2016) provide an overview of state-of-the-art approaches for weather routing, noting that all approaches present significant benefits and deficiencies and the weather routing topic remains heavily researched. Veneti et al. (2017) suggest a detailed framework for weather routing based on a shortest-path algorithm, where vessel's FOC is a static approximation based on first principles. Accordingly, Vettor and Guedes Soares (2016) combine a first-principle ship response estimation with a genetic algorithm approach for the derivation of an optimal vessel route. Lin et al. (2013) propose a combination of a mathematical model that predicts ship performance with a three-dimensional modified isochrone (3DMI) method for weather routing. Similarly, Roh (2013) suggests a framework for weather routing based on first principles resistance-based FOC estimation and the isochrone method for weather routing. R. Lu et al. (2015) propose another semi-empirical method for the prediction of operational performance of ships based on modelling ship' resistance. The ship's operational performance is then modelled, taking into consideration the weather and relevant sea state and utilised to optimise the ship's route.

Moreover, Armstrong and Banks (2015) provide an integrated overview of energy efficiency improvements for vessel sailing, highlighting the necessity of a set of well-defined KPIs to simplify performance tracking. Acomi and Acomi (2014) suggest the use of Energy Efficiency Operational Indicator (EEOI) as a KPI for the amelioration of vessel sailing FOC efficiency.

From the above, it can be deduced that weather routing is an active research field with

multiple different approaches being realised concurrently. However, up to present, most studies focus on first-principle methods for the estimation of a vessel's performance. Due to this, the approaches suggested do not accurately take into account a vessel's current condition and sailing peculiarities, such as hull fouling, degraded engine performance. More so, the underlying FOC modelling often cannot provide accurate predictions in the presence of adverse weather conditions.

3.5 Identified gaps

This chapter includes the critical review of publications pertaining to maritime condition monitoring, prediction of a vessel's FOC, and weather routing. Through this review, a number of research gaps have been identified along with relevant research directions. These will play a pivotal role in the justification of the framework proposed in Chapter 4 and help establish its inherent novelty.

The key gaps identified are discussed below, categorised by application, i.e. maritime condition monitoring, FOC estimation, and weather routing. Furthermore, some key remarks are discussed, concluding the conducted review.

Maritime condition monitoring

Reviewing the pertinent literature, it is evident that CM remains an active research field. This statement is even stronger in the context of the maritime sector, where end-to-end solutions are rarely available and preventive maintenance is often the preferred way forward. Due to this, peculiarities of the industry need to be considered to propose solutions that can transcend the state-of-art and become the state-of-practice.

It is important to note that no one-size-fits-all modelling approach exist as different monitoring goals and system architectures across various industries lead to different conclusions regarding optimal modelling approaches. Therefore, in order to obtain the full benefits of condition monitoring, fault identification, localisation, and RUL estimation need to be performed. This is clearly a step-by-step process where simpler

solutions need to be developed and successfully implemented before the introduction of advanced techniques with increased demands at both the modelling and the deployment stages.

Reviewing the development of other industries, a frequent first attempt at CM is anomaly detection as the data requirements are limited compared to other approaches. CM has not yet been thoroughly examined and applied in the maritime industry.

FOC prediction

As DAQ systems become commonplace, FOC prediction in the context of the maritime sector remains an active research field as more data become available, leading to more accurate and robust modelling. Past literature has claimed that a number of regression techniques have performed best on this task based on different datasets and data acquisition methods but no systematic investigation has been performed so far. Specifically, it has been noticed that many studies reaching contradicting conclusions have been published, without a clear indication as to which method should be applied in order to achieve optimal modelling results. Moreover, FOC prediction models have been investigated on their own but the investigation of applications based on such models (e.g. performance monitoring) remains scarce. Especially the use of FOC models as proxy for either the condition or performance status of a vessel retains untapped potential and should be further investigated.

Weather routing

Vessel weather routing remains an active research field as there are significant differences to the predictive systems used in other applications, e.g. vehicle routing. Furthermore, the uncertainty and the short time-frame of the available data poses additional challenges that need to be overcome before such a solution can be deployed. Vessel weather routing often applied first-principle vessel models but these models often cannot take into account performance changes due to fouling or adverse weather. Hence, Data-driven approaches have not been investigated in depth for route optimisation

purposes.

General remarks

As a general comment stemming from the review of all studies that relate to data-driven applications, thorough and accurate data pre-processing is required to achieve accurate and robust modelling results. Moreover, while individual tools or frameworks that tackle specific problems exist or are in active development, the integration of such tools in order to enhance the operational efficiency of vessels in a more holistic approach is scarce.

Furthermore, while both vessel condition and vessel performance monitoring have been considered in past literature, a holistic condition and performance monitoring approach has not yet been investigated. Finally, weather routing that encompasses vessel condition and performance information should be investigated as a direction towards more accurate results.

3.6 Chapter summary

In this chapter a thorough review of existing literature was presented. Initially an overview of maintenance strategies was presented. Following that, focus was shifted to CM, first discussing CM applications in the maritime industry and then critically reviewing relevant applications in other industries. Additionally, data-driven methods for the prediction of FOC were reviewed, followed by methods for weather routing. Furthermore, ancillary to the above, an high-level description of common modelling approaches is provided. Finally, by critically identifying gaps in the existing literature, the thesis framework is proposed and thoroughly presented in the next chapter.

Chapter 4

Methodology & Modelling

Based on the gaps identified through the critical literature review conducted as part of Chapter 3, this chapter aims to present and elaborate on the proposed modelling framework oriented towards the improvement of a vessel's operational efficiency. This framework utilises ship raw data to monitor the condition of ship systems, identify optimal Fuel Oil Consumption (FOC) prediction modelling approaches, monitor the sailing performance of a vessel based on its FOC, and provide optimal routing decision support.

Initially, Section 4.1 provides an overview of the framework described within this thesis, before elaborating separately on its each methodological element. Section 4.2 refers to the process followed for the data collection of all types of data required. Section 4.3 describes the steps applied to different data subsets for them to be modified in the necessary ways before being used for model training. Section 4.4 discusses best practices for data-driven model building, as these remain common for all models developed. Sections 4.5 – 4.9 elaborate on the derivation and development of individual models along with their integration. These sections include the development of the novel methodologies that form the backbone of this thesis, namely the condition monitoring methodology, the FOC modelling comparative methodology, and the weather routing methodology. Finally, Section 4.10 provides some concluding remarks on the proposed

framework and an overall summary of this Chapter.

4.1 Overview of proposed methodology framework

This section presents the overall data-driven vessel operational efficiency enhancement framework, proposed within this thesis. More specifically, the framework aims to present an integrated method that applies advanced machine learning techniques for operational efficiency enhancement through: (a) a novel condition monitoring methodology; (b) a novel FOC modelling comparative methodology; (c) a FOC-based performance monitoring methodology; and (d) an optimal routing decision support methodology.

The methodology initially provides a systematic approach to data collection and preprocessing in order to ensure robustness and repeatability. Following that, while not explicitly being part of the framework, best practices relevant to model training are discussed as these constitute a commonality between different models developed for different purposes. As the next step, models that predict a vessel's mechanical condition and FOC-based performance for monitoring purposes are developed. In the case of condition monitoring of the propulsion system, a One Class Support Vector Machine (OC-SVM) is developed to identify system states that deviate from normal system operation, i.e. anomaly detection. Accordingly, in the case of predicting a vessel's performance based on its FOC, a methodology for the evaluation of individual regression models is proposed and developed, aiming to identify some of the best-performing architectures for this task, under a number of scenarios. Based on the results of this comparison, a suitable model is derived, able to predict a vessel's FOC over a specific route, given a number of ambient weather and voyage-specific operational parameters. Comparing the results of this model to it's actual FOC over the same route and given the same parameters, its current performance deviation can be identified. As an extension to this task, a number of relevant Key Performance Indicators (KPIs) are proposed. Last but not least, this model combined with a novel heuristic driving Dijkstra's algorithm is used to provide optimal routing decision support, taking all the

above into consideration. While Figure 4.1 provides an visual overview of the proposed methodological framework, every element of the framework is elaborated in the Sections that follow.

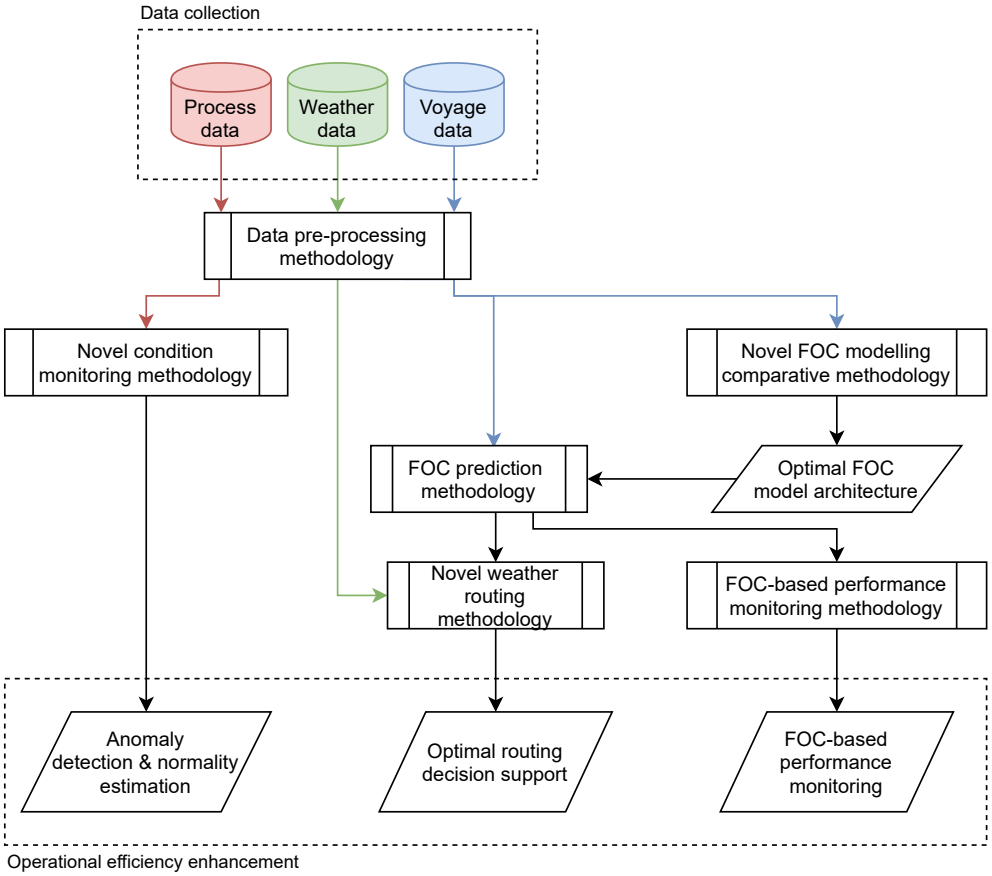

Figure 4.1: Visual representation of the overall proposed methodological framework.

4.2 Data collection process

The data required for the training of the machinery condition monitoring and vessel FOC prediction models elaborated in this Chapter can either be acquired through an Automated Data Logging & Monitoring (ADLM) system, or through the processing of noon reports depending on relevant availability. While noon reports are traditionally human-generated documents that record voyage data at a daily frequency, in some of the datasets used in the case studies of this thesis, the reports provided were augmented

with information concerning the condition of key machinery or recorded at a higher frequency. While these reports are technically not noon reports, they will be referred to as such, to differentiate them from ADLM sources. Compared to parsing noon-reports, ADLM systems provide higher-frequency data of increased accuracy, albeit at an elevated cost. The term voyage data refer to data that describe a vessel's current or historical voyages. Such data include encountered weather conditions, vessel speed, drafts, Main Engine (M/E) speed and FOC. Accordingly, the term process data refers to measurements obtained from the M/E or other critical pieces of machinery. These data do not directly relate to the vessel's voyage but that can be used to monitor machinery condition.

Regarding the data requirements of the condition monitoring methodology, an additional need arises. As will be discussed in more depth in Section 4.5 where the condition monitoring methodology is presented and analysed, data points used in this methodology for model training purposes need to have their veracity ensured and to conform to normal operating conditions. This requirement establishes a new, intermediate, step in the data collection process for this methodology. This dual verification is performed in tandem through data analysis and through the acknowledgement of expert judgement.

For weather routing, a second dataset containing ocean analysis and forecast data (summarised as "weather data" in Figure 4.1) for the sailing region is required. This is obtained through weather providers, such as Copernicus Marine Environment Monitoring Service (CMEMS), and depending on provider and attribute characteristics come with different spatial and temporal resolutions. For example, CMEMS global wave forecast has a maximum spatial resolution of 1/12 degree and a 3-hourly temporal resolution, providing 5 days of forecast.

Due to the existence of potential measuring anomalies, lack of data, and undesirable data points (e.g. engine transients), pre-processing follows the data acquisition phase for both datasets.

4.3 Data pre-processing methodology

In the case of data required for the training of the machinery anomaly detection and vessel FOC prediction models, engine transients and recording anomalies are identified and rejected. Regarding the weather forecast dataset, instances with missing information are additionally imputed to obtain an uninterrupted dataset. Imputation is not performed for the condition and performance models, as additional dataset noise and uncertainty would be introduced without a tangible benefit in regards to model accuracy. Following that, existing dataset features are manipulated, utilising domain knowledge, to produce new features than can increase a model's accuracy. Finally, all features are standardised to ensure that all features contribute equally to a model's output. In Figure 4.2, the overall methodology relating to this section is presented, elaborating how different approaches apply to the three kinds of data inputs considered, namely: voyage data, process data, and weather data.

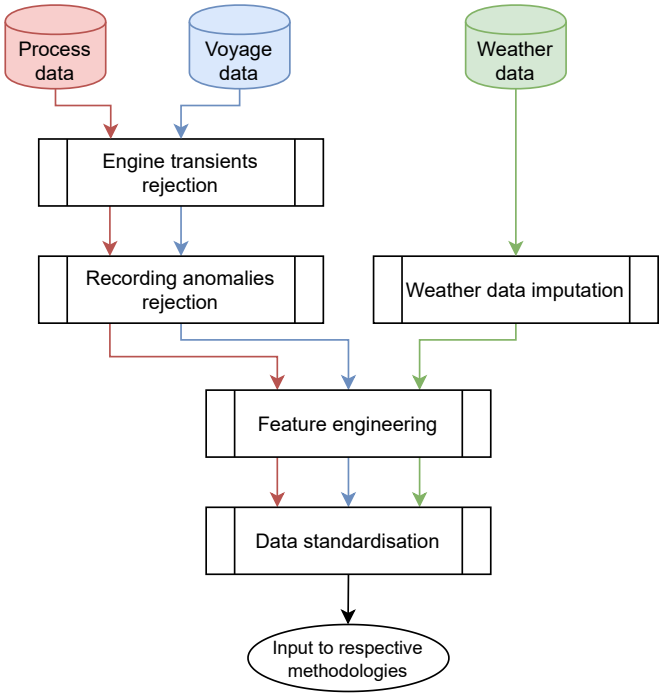

Figure 4.2: Visual representation of the proposed pre-processing methodology.

4.3.1 Engine transients rejection

The goal of the condition monitoring methodology that will be elaborated in Section 4.5 is to identify engine anomalies whilst the engine is operational, i.e. at real-time or close to it. However, detecting anomalies at a transient engine state exhibits increased complexity (e.g., the requirement to treat data points strictly as a time-series arises) without significant added benefits due to the system degradation being a monotonous phenomenon. For this reason, transient operation is detected and the relevant data points are discarded.

Accordingly, the derived FOC-based performance monitoring, and optimal routing decision support methodologies aim to provide actionable information that can be used to enhance a vessel's operational efficiency. As the underlying phenomena (e.g. hull fouling, weather deterioration) exhibit gradual transformations, the decision making itself is performed considering a time-scale in the order of hours, with the predictive models needing to provide performance and FOC predictions at discretised segments that, depending on the spatial resolution of the available weather forecast, often span several hours. Based on that, only steady-state (i.e. continuous) historical sailing data are used for its training.

In this respect, M/E Original Equipment Manufacturers (OEMs) provide a minimum engine speed for continuous operation (MAN B&W Diesel A/S, 2004), usually at $15 - 25\%$ of the engine's nominal maximum continuous (L_1) speed. Any observations corresponding to measured speed below that threshold are then rejected as an engine transient or manoeuvring. Additionally, observations where the engine power varies by more than 5% hourly are also discarded as the FOC is only trained on steady-state data due to the low temporal resolution of the weather forecast provider (Tsitsilonis & Theotokatos, 2018).

4.3.2 Recording anomalies rejection

Voyage and process data acquisition may potentially introduce inconsistent and/or faulty data entries (e.g. due to sensor anomalies, or human error) that need to be

identified and rejected at this stage.

Applying the Central Limit Theorem (CLT) given that sensor data are aggregated every few seconds or minutes, data can be assumed to follow a normal distribution. CLT establishes that given a number of independent, random variables, the distribution of their normalised sum can be approximated by the normal distribution, even if the original variables do not follow that distribution.

Therefore, 99.7% of normal data can be assumed to lie within $\mu \pm 3\sigma$, where μ corresponds to the mean value of each attribute and σ to its standard deviation. Hence, observations outwith this range can be rejected as anomalous without affecting the vast majority of normal points.

At the same time, operational and sailing data are scanned and observations where missing features exist either due to sensor malfunctions or due to a piece of machinery not operating at a given point are discarded. Missing data points are usually denoted in the dataset as *Not a Number* (NaN)).

4.3.3 Weather data imputation

As previously mentioned, in the case of process and voyage data, observations where missing features exist get discarded to avoid tainting the training dataset. However, in the case of weather data, data points (i.e. weather information) at every grid point and at every time-step are necessary input to the routing optimisation algorithm. Conversely, lack of weather information at a data point denotes that this grid point is not a feasible route point (e.g. due to it being part of land) and that a detour has to be applied.

Therefore, missing weather data values have to be imputed based on the information contained in the available weather data. Multiple Imputation (MI) represents a regression-based imputation approach that provides increased accuracy while controlling the latent bias of the process (Azur et al., 2011). For the weather data imputation required as part of the pre-processing methodology, Multiple Imputation by

Chained Equations (MICE) algorithm was applied as it constitutes one of the most promising implementations of MI. MICE algorithm can be summarised as an imputation approach that fits a number of regression models to the dataset in order to infer missing values (Shah et al., 2014). A brief description of the steps required for the application of MICE algorithm is as follows:

1. Replace all missing values with the mean value of their attribute as an initialisation step. Keep track of instances where this has happened.
2. Revert the missing values of a single attribute to missing.
3. Fit a linear regression model to predict missing values of that attribute based on some (or all) other attributes.
4. Repeat steps 2 and 3 until no attributes have missing points.
5. Repeat steps 1 – 4 for a predetermined number of cycles.

4.3.4 Feature engineering

Feature engineering is the process of applying data mining techniques to a raw dataset in order to extract insights that can be used to increase the performance of machine learning models (Domingos, 2012). Depending on the methodology and relevant data sources, different feature engineering approaches have been applied. These are described below, on a methodology-by-methodology basis.

4.3.4.1 Condition monitoring methodology, FOC modelling comparative methodology & FOC prediction methodology

Given domain knowledge of the available parameters for FOC modelling, transformations can be performed to engineer new features that better capture the information contained in the raw dataset.

For example, forward and aft draft observations can be transformed into draft amidships and trim features, as these features can be, potentially, more accurate predictors for

the FOC of the vessel. Accordingly, in cases where flow meters are installed in both the inlet and return lines, the difference of the two measurements can be computed to obtain a single target variable for the model.

Similarly, in the case of condition monitoring data, measurements such as pressure and temperature before/after a location can be combined in a single feature (e.g. pressure drop, or temperature differential) that better represent the relevant information.

Furthermore, in order to ensure consistency between data points acquired under different ambient conditions, measured parameters are corrected to International Organization for Standardization (ISO) ambient conditions (MAN B&W Diesel A/S, 2004).

4.3.4.2 Weather routing methodology

All ocean analysis and forecast data follow a space-fixed frame of reference, i.e. are either provided in the form of a northward and eastward component or in that of an angle and a magnitude. As the effect of the weather elements depends on the current course of the ship, these measurements are transformed into a body-fixed frame of reference.

To transform measurements that are provided in a northward (nw) and eastward (ew) component into the vessel body-fixed frame of reference sailing at a bearing br and obtain a longitudinal (lg) and transverse (tr) component, the following formulas can be used (Figure 4.3):

$$lg = nw\ \cos(br) + ew\ \sin(br) \tag{4.1a}$$

$$tr = |ew\ \cos(br)| + nw\ \sin(br) \tag{4.1b}$$

In the case where the measurements are described by a magnitude M and an angle α, they can be transformed into the vessel body-fixed frame of reference sailing at a bearing br and obtain longitudinal (lg) and transverse (tr) components by applying the

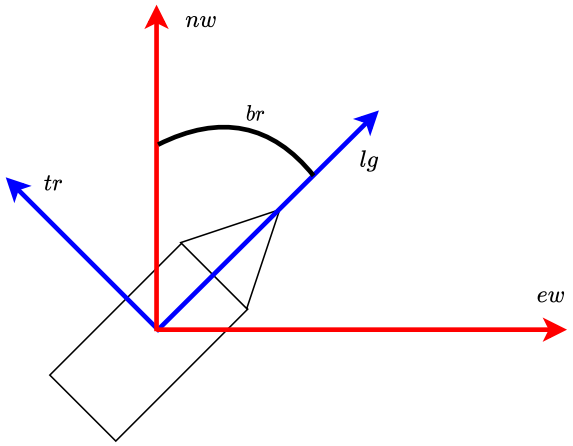

Figure 4.3: Visual representation of weather forces acting on vessel in a space-fixed frame of reference compared to a body-fixed.

following formulas (Figure 4.4):

$$lg = M\ \cos(\alpha - br) \tag{4.2a}$$

$$tr = |M\ \sin(\alpha - br)| \tag{4.2b}$$

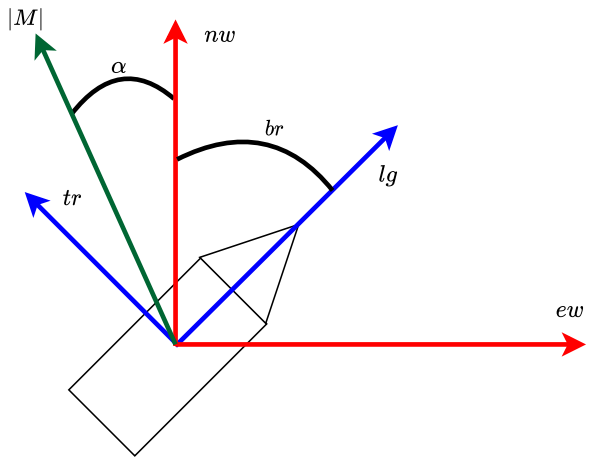

Figure 4.4: Visual representation of weather forces described in a magnitude, angle tuple acting on vessel in a space-fixed frame of reference compared to a body-fixed.

4.3.5 Data standardisation

All numerical attributes in the dataset are standardised by removing the mean and scaling to unit variance. Therefore, for a numerical attribute x, a standardised attribute x' is produced by

$$x' = \frac{x - \mu}{\sigma} \tag{4.3}$$

All attributes are standardised so that all attributes can contribute equally to the objective function that is used for model training.

4.4 Model training best practices

This section elaborates on some key model training principles, such as k-folding, and hyperparameter optimisation, that apply to all model architectures that are developed in the following sections. Due to this, these principles are all discussed in depth at this point and then only briefly discussed in the next sections instead of being repeated at every point.

4.4.1 K-folding

Training dataset through K-folds cross-validation is used both for model training (i.e. deriving optimal model parameters) and hyperparameter optimisation (i.e. deriving optimal hyperparameter values). Model parameters are learnt during model training and depend on the training dataset peculiarities. Contrarily, model hyperparameters cannot be learnt during training but instead are selected by the user in order to optimise a selected metric (e.g. accuracy). Hence, while model parameters vary for distinct training data sets of the same application, hyperparameters tend to be common for similar models.

K-fold cross validation works by splitting the training dataset X into K roughly equal parts $X_1, X_2, \ldots, X_K$ and using $K - 1$ of them for training and 1 for validation, going

through a for-loop to ensure that all K possible combinations are evaluated. Through K-fold cross validation, the generalisation capabilities of a model can be established as the model is trained in K different scenarios and the average performance is evaluated. A visual example of this procedure is presented in Figure 4.5. Additionally, through the introduction of a second for-loop, different hyperparameter tuples can be evaluated in order to identify which one optimises model performance. This is presented in pseudo-code in Algorithm 1. Therefore, for each hyperparameter combination, several results are obtained and averaged.

Algorithm 1 Model training and hyperparameter optimisation using K-folding

Require: X, X_{abn}, K, a set of n hyperparameter tuples $h_i, i \in (1, 2, \ldots, n)$
1: Divide data X into K roughly equal parts
2: **for** i = 1, 2, ..., n **do**
3: $Hyps \leftarrow h_i$
4: **for** k=1, 2, ..., K **do**
5: $TrainSet \leftarrow X \setminus X_k$ (set subtraction)
6: $ValidSet \leftarrow X_k \cup X_{abn}$
7: Train model $M_{(k,i)}$ using $TrainSet$ and $Hyps$
8: Evaluate model $M_{(k,i)}$ quality $MCC_{(k,i)}$ on $ValidSet$
9: **end for**
10: Calculate mean model quality $MCC_{(\mu,i)}$ (Eq. (4.11)) by averaging $MCC_{(k,i)} \forall\, k$
11: **end for**
12: Obtain $h^* = \underset{h_i}{\operatorname{argmax}} (MCC_{\mu,i})$ and use it to train final model M^* on whole X
13: **return** M^*

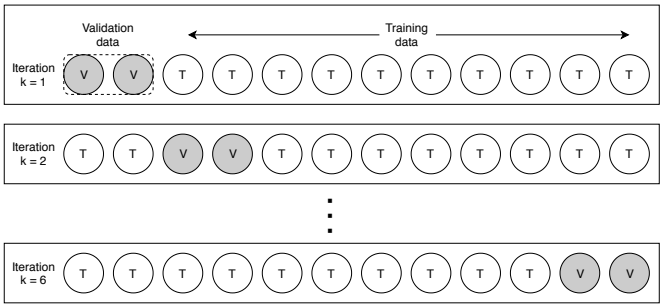

Figure 4.5: Visual representation of k-folding for 12 data points and k=6. In this example, the dataset is split in six subsets, and for every value of k, a different subset is selected for validation whilst the rest are used for model training. This helps prove the robustness of the model and its hyperparameters.

Finally, a testing sub-dataset is used to evaluate the generalisation capabilities of the

final selected model (Hastie et al., 2009) using a previously unseen dataset, in order to yield a more accurate, real-world usage scenario.

4.4.2 Hyperparameter optimisation

As mentioned previously, model hyperparameters refer to parameters that can arbitrarily be set before model training and do not directly depend on the training data. As the optimal hyperparameter values cannot be known *a priori*, a number of optimisation techniques can be applied to identify the best hyperparameter values for each model. A naïve method to do so would imply building a grid containing all possible combinations of selected hyperparameters and exhaustively evaluating each to select the best combination. However, this carries a significant cost due to the sheer number of combinations that are evaluated (especially in the case of multiple tuneable hyperparameters per model). Another approach is to employ a random search implementation; there, all hyperparameter ranges are randomly sampled – usually producing more accurate results given a predefined number of draws (Bergstra et al., 2011). In some cases, more advanced optimisation methods, such as Bayesian optimisation, are applied. This holds particularly true in cases where model training is computationally expensive and the number of optimisation iterations needs to be minimised.

4.5 Novel condition monitoring methodology

This section presents the development of a self-learning model that can estimate whether a given data point corresponds to a reference (nominal) condition considered during model training. As such, a self-learning model can be trained without the need of obtaining data corresponding to "faulty" conditions. This allows the detection of anomalous operating conditions of machinery and the estimation of their subsequent mechanical condition normality. A flowchart of the proposed approach for machinery anomaly detection is presented in Figure 4.6.

Compared with other relevant methodologies available in pertinent literature, the novel characteristic of this methodology concerns the modelling of a condition-estimation

system using only "normal" (i.e. nominal) data and the transformation of its output to a time-dependent (dynamic) normality metric without the explicit need for any additional information. Additionally, the developed methodology can be applied for the establishment of relevant measurement thresholds when such values are not readily available.

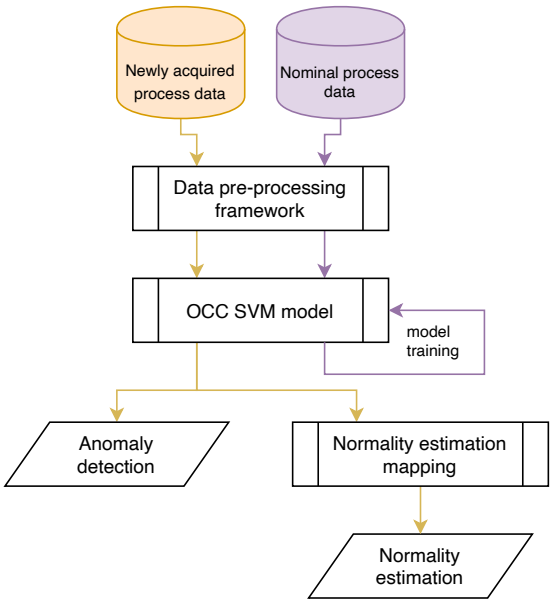

Figure 4.6: Visual representation of proposed condition monitoring methodology.

A one-class Support Vector Machine (SVM) classifier (Cortes & Vapnik, 1995) is used, as it is one the most commonly chosen algorithms for One Class Classification (OCC) (Khan & Madden, 2010). Compared to other machine learning tools, SVMs offer superior generalisation capabilities (Widodo & Yang, 2007). This is a OCC algorithm that learns a decision boundary using only "normal" data points and then testing the likelihood of a test instance being within the boundary of the learnt model. Schölkopf et al. (2001), Schölkopf et al. (2000), Schölkopf et al. (1999) present a method of creating an OCC by implementing a suitable separating hyperplane of the form:

$$w^T x + b = 0 \tag{4.4}$$

where w is the weight vector, always normal to the hyperplane, x is the input vector and

b is the bias term. This hyperplane aims to separate the surface region containing data from the region containing no data. This is achieved by "constructing a hyper-plane which is maximally distant from origin, with all data points lying on the opposite side from the origin and such that the margin is positive" (Khan & Madden, 2010).

In their ν-SVM implementation (Schölkopf et al., 2000), SVMs utilise the parameter ν as a degree of freedom in the trade-off between a large margin and a small training error. Accordingly, the parameter ν is "an upper bound on the fraction of training margin errors and lower bound on the fraction of support vectors" (Wu & Srihari, 2003). In more practical terms, $\nu \in (0, 1]$ represents both an upper limit on the number of misclassifications in the training dataset (at the cost of a possibly smaller margin) and a lower limit in the number of training samples used as support vectors (C.-C. Chang & Lin, 2001).

Following the formulation suggested by Schölkopf et al. (1999), the objective function of a SVM OCC can be described as:

$$\begin{gathered} \min_{w,\xi_i,\rho} \left[\frac{1}{2} \|w\|^2 + \frac{1}{\nu\,\ell} \sum_{i=1}^{l} \xi_i - \rho \right] \\ \text{subject to: } (w \cdot \Phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0 \end{gathered} \tag{4.5}$$

where ℓ is the number of observations, ρ represents the offset, and ξ_i are slack variables introduced to allow some points to lie within the margin in order to avoid overfitting.

Solving this minimisation problem using Lagrange multipliers, the decision function rule for a data point $\boldsymbol{x}$ then becomes:

$$f(x) = \operatorname{sgn}\left(\sum_i \alpha_i \kappa(\boldsymbol{x_i}, \boldsymbol{x}) - \rho \right) \tag{4.6}$$

Coefficients a_i can be found as solutions to the dual problem:

$$\begin{aligned} \min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) \\ \text{subject to: } 0 \leq \alpha_i \leq \frac{1}{\nu \ell}, \sum_i \alpha_i = 1 \end{aligned} \tag{4.7}$$

where κ refers to the kernel function that will be elaborated in subsection 4.5.1.

In order to calculate the offset ρ, we can exploit the fact that for any α_i that is not at the upper or lower bound, the corresponding pattern $\boldsymbol{x_i}$ satisfies:

$$\rho = (w \cdot \Phi(x_i)) = \sum_j \alpha_j \kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) \tag{4.8}$$

4.5.1 Kernel functions

A key selection point in a SVM implementation concerns the kernel function κ, that for two vectors $\boldsymbol{u}, \boldsymbol{v}$ takes the form:

$$\kappa\left(\boldsymbol{u}, \boldsymbol{v}\right) = \Phi(\boldsymbol{u})^T \Phi(\boldsymbol{v}) \tag{4.9}$$

Kernels operate as a similarity function, offering a gauge of similarity between two inputs and, especially in the case of SVMs, as a transformation that helps to linearly separate linearly inseparable data. In that case, the kernel function offers a map of the originally inseparable data to a higher-dimensional space where they can be linearly separable. While multiple kernel functions can be applied for model training, Radial Basis Function (RBF) is, in practice, considered to work well as a SVM kernel and is usually a reasonable first choice (Hsu et al., 2010). As indicated in Equation (4.10), a significant advantage of RBF compared to other kernel functions is that it can be easily calibrated as it only depends on one parameter, $\gamma \in (0, \infty)$. A reasonable range and search spacing for γ when used as a SVM kernel, is $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^3$ (Hsu et al., 2010). Parameter γ controls the region that a single training sample can affect, with a

small value of γ increasing the size of this region; and conversely. In other words, a large value of γ leads towards over-fitting in the Bias-Variance trade-off (Geman et al., 1992) whereas selecting a small value risks creating a model too constrained to efficiently capture the complexity of the training dataset. For two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, RBF function $\kappa(\boldsymbol{u}, \boldsymbol{v})$ is determined by Equation (4.10).

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\gamma \|\boldsymbol{u} - \boldsymbol{v}\|^2\right) \tag{4.10}$$

4.5.2 Hyperparameter optimisation

Recapping the above, the performance of a one-class SVM classifier utilising RBF as kernel function and given a specific training dataset depends on the two hyperparameters: γ and ν. The optimal values of these hyperparameters are selected through random search (Bergstra et al., 2011); a predefined number of ν and γ values are randomly selected from their relevant search space and fed as hyperparameter tuples to the model optimiser.

Models are trained for each hyperparameter tuple in the two-dimensional grid using the training sub-dataset through cross-validation. The hyperparameter tuple h^* that, on average, achieves the best results (lower error) , are finally selected and a model M^* is trained based on the whole dataset and h^*.

As previously mentioned, K-fold cross validation is implemented in order to safeguard a model's generalisation. However, in the case of OCC, only having validation data points belonging to one class does not suffice to ensure adequate generalisation. At the same time, access to abnormal data points at the validation stage cannot be merely assumed while developing a methodology, as in that case, developing a multi-class classifier with imbalanced data sets (e.g. Chawla et al. (2004) and Van Hulse et al. (2007)) would be more appropriate. For this reason, a number of abnormal data points are algorithmically simulated without any requirements for a priori knowledge of the physical system. These points are generated by adding 4 or 5 standard deviations to

the actual values, therefore, the probability that they are normal is in the range of 6×10^{-5} to 6×10^{-7}. The algorithm implemented to produce this simulated abnormal dataset is shown in Algorithm 2. Having derived the abnormal validation dataset, this is appended to the "normal" one, i.e. the one derived through K-folding and the two are concatenated.

Algorithm 2 Algorithmic derivation of abnormal points in array X_{abn}

Require: X

$n_{attr} \leftarrow$ number of attributes in X
Initialise X_{abn} with $4 \times n_{attr}$ randomly selected observations from X
for i = 1, 2, ..., n_{attr} **do**
    $attr_{max} \leftarrow \max(X[i])$
    $attr_{min} \leftarrow \min(X[i])$
    $attr_{std} \leftarrow \text{std}(X[i])$
    $X_{abn}[4i-3, i] \leftarrow attr_{max} + 4 \times attr_{std}$
    $X_{abn}[4i-2, i] \leftarrow attr_{min} - 4 \times attr_{std}$
    $X_{abn}[4i-1, i] \leftarrow attr_{max} + 5 \times attr_{std}$
    $X_{abn}[4i, i] \leftarrow attr_{min} - 5 \times attr_{std}$
end for
return X_{abn}

In order to evaluate the measure of the quality of the derived model, the Matthews Correlation Coefficient (MCC) of the validation dataset is calculated for each hyper-parameter tuple. MCC returns the correlation coefficient between actual and predicted binary classifications and returns values in the range $[-1, +1]$. MCC can be calculated by using the following equation (Matthews, 1975):

$$\begin{aligned} \text{MCC} &= \frac{TP/N - S \times P}{\sqrt{P \times S\,(1-S)\,(1-P)}} \\ \text{N} &= TN + TP + FN + FP \\ \text{S} &= \frac{TP + FN}{N} \\ \text{P} &= \frac{TP + FP}{N} \end{aligned} \tag{4.11}$$

where TP corresponds to true positives, TN to true negatives, FP to false positives and FN to false negatives.

4.5.3 Normality estimation mapping

Following the previous steps, a hyperplane that encompasses the original normal points is derived in order to label accordingly new points as either normal or abnormal. At the same time, the distance between the hyperplane and each new point can be used as a normality metric. The further a point is from the hyperplane towards the normal side, the normality of the system examined increases, and vice versa. This also presents an inherent limitation of one-class models, i.e. their ability to only identify abnormalities at system level, without the ability to examine individual components for maintenance optimisation.

This can then be mapped into a 0-100% semi-qualitative range in order to simplify evaluation and further processing. The normality assigned to these points can be altered depending on specific case requirements. This mapping is developed by deriving a function that performs the non-linear mapping of points presented in Table 4.1.

Table 4.1: Normality mapping control points

Control point description	normality assigned (%)
Data point in training set that yields maximum (positive) distance to hyperplane	100
Data point in training set that yields minimum (positive) distance to hyperplane	user-selectable α
Data point in simulated abnormal set that yields minimum (negative) distance to hyperplane	0

Additionally, as every data point is evaluated independently, an Exponential Weighted Moving Average (EWMA) filter is implemented to smooth-out unnecessary kinks and make the evaluation of the overall degradation easier. EWMA for a series Y can be calculated recursively by using the following equation (Croarkin et al., 2018):

$$S_t = \begin{cases} Y_1, & t = 1 \\ a \cdot Y_t + (1 - \alpha) \cdot S_{t-1}, & t > 1 \end{cases} \tag{4.12}$$

where $t = 1$ refers to the first point, with t increasing for each subsequent point and parameter α can be calculated from the Centre Of Mass (COM) property through the

following equation:

$$\alpha = 1/\left(1 + COM\right) \tag{4.13}$$

4.6 Novel FOC modelling comparative methodology

The methodology elaborated in this section consists of the following steps: a) the development and implementation of multiple models following different modelling approaches; b) the optimisation of the hyperparameters of these models; and c) the comparison of these models to identify the modelling techniques that offer the best performance.

A visual representation of the developed methodology is presented in Figure 4.7, illustrating all suggested modules and their relevant interconnections.

4.6.1 Modelling approaches

All modelling approaches presented below are methodologies related to regression analysis, as regression analysis provides a means for the prediction of a dependent variable (i.e. FOC) given a number of known, independent variables (e.g. weather conditions, voyage information). Regression models may be derived with a varying level of complexity and consequently accuracy of results. Therefore, possible methods span a wide range of options, from closed-form linear models to deep (i.e. multi-layered) neural networks (Bishop, 2006; Russell & Norvig, 2010).

4.6.1.1 Parametric versus non-parametric modelling

Modelling approaches can be split into two major categories: parametric and non-parametric. Parametric models assume some finite set of parameters θ that are obtained from the training set during the learning phase (Bishop, 2006). Following that phase, the training set is discarded and any future predictions x are independent of the observed dataset D so that:

$$P\left(x \mid \theta, D\right) = P\left(x \mid \theta\right) \tag{4.14}$$

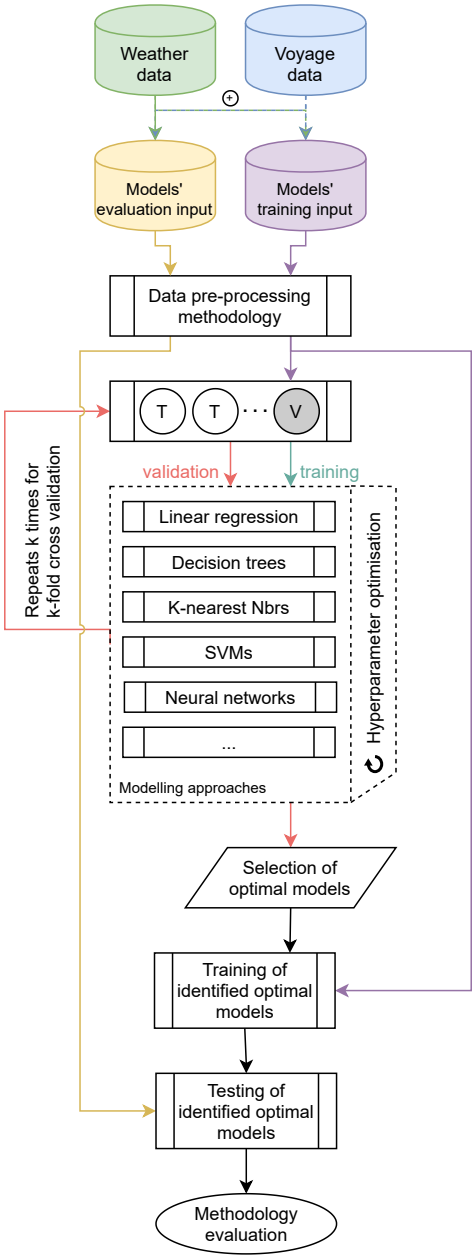

Figure 4.7: Visual representation of the suggested FOC modelling comparative methodology.

In other words, θ is assumed to capture all variance contained in the dataset D (Clarke et al., 2009). Therefore, even if the complexity of a dataset is unbounded (potentially infinite), the complexity of the model is bounded (Russell & Norvig, 2010). Models such as linear regression, Artificial Neural Networks (ANNs), and Support Vector Reggressors (SVRs) with a linear kernel are parametric models.

In contrast to that, non-parametric models assume that the dataset distribution cannot be defined using any finite number of parameters. For this reason, in non-parametric models, training data, or at least a subset of them, are kept and utilised during the prediction phase (Bishop, 2006). Therefore, the amount of information that θ can capture grows with the number of training data points in dataset D. Decision tree regressors, random forest regressors and SVRs with a RBF kernel are considered non-parametric as the number of parameters grows with the size of D.

Following the above, non-parametric modelling approach can potentially provide higher-performance models due to a reduced number of parameter assumptions. However more training data are required and the computation cost is increased.

Finding the optimal model-derivation methodology is non-trivial as this is affected, among others, by the quantity and quality of available data, and the nature (and also complexity) of the problem at hand.

4.6.1.2 Multiple linear regression

Linear Regression (LR), a parametric model, constitutes the simplest regression algorithm, involving a linear combination of the input variables $\mathbf{x} = (x_1, \ldots, x_D)$ (Bishop, 2006):

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1\, x_1 + \ldots + w_D\, x_D = w_0 + \sum_{j=1}^{D} w_j\, x_j \tag{4.15}$$

Parameters w_j, $j \in (0, \ldots, D)$ of Equation 4.15 can then be estimated using a Least Squares (LS) approach as:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{D} (w_j \, x_{ij}) \right)^2 \right\} \tag{4.16}$$

Multiple Linear Regression (MLR) constitutes an extension of LR, in cases where $D > 1$ (Hastie et al., 2009). LR and MLR models are often used as a baseline, against which the performance of other models is evaluated.

4.6.1.3 Ridge & LASSO regression

Ridge Regression (RR) follows the concept of MLR but instead of using the parameters w_i derived through LS, in RR these parameters are shrunk by imposing a penalty on the square of each parameter (Hastie et al., 2009). In this case, Equation 4.16 obtains an additional regularisation parameter and becomes:

$$\hat{w}_{\text{RR}} = \underset{w}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{D} (w_j \, x_{ij}) \right)^2 + \lambda \sum_{j=1}^{D} w_j^2 \right\} \tag{4.17}$$

where hyperparameter $\lambda > 0$ is a user-selectable parameter that controls the amount of shrinkage. This shrinkage helps avoid overfitting the training dataset.

Least Absolute Shrinkage and Selection Operator (LASSO) is another shrinkage method, similar to RR, with the main difference being that the penalty is imposed on the absolute value of each parameter instead of their squares. Therefore, parameters w_i can now be predicted as

$$\hat{w}_{\text{LASSO}} = \underset{w}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{D} (w_j \, x_{ij}) \right)^2 + \lambda \sum_{j=1}^{D} |w_j| \right\} \tag{4.18}$$

An extended version of LASSO and RR are Elastic Nets, where both absolute-value and squared regularisations are implemented concurrently, with the regularisation term

of Equations 4.17, 4.18 which become

$$\lambda_1 \sum_{j=1}^{D} |w_j| + \lambda_2 \sum_{j=1}^{D} w_j^2 \tag{4.19}$$

In the the Scikit-learn implementation (Pedregosa et al., 2011), hyperparameters α and $\lambda 1_{ratio}$ are used instead. The following equations transform λ_1 and λ_2 to α and $\lambda 1_{ratio}$:

$$\alpha = \lambda_1 + \lambda_2 \tag{4.20a}$$

$$\lambda 1_{ratio} = \frac{\lambda_1}{\lambda_1 + \lambda_2} \tag{4.20b}$$

4.6.1.4 Decision tree regressors

Decision Tree Regressors (DTRs) are a non-parametric, regression method. DTR models partition the feature space into rectangles and learn a simple (e.g. constant) model in each of those (Hastie et al., 2009).

DTRs do not produce a continuous output in the traditional sense. Instead, these models are trained on a training set whose outputs lie on a continuous range. Their output ends up being the mean value of the training set observations that reside in the same node.

One of the most common methods for tree-based regression is Classification And Regression Trees (CART) (Breiman et al., 1984). In this case, the original feature space is split into two regions, selecting the split point and dependent variable (feature) to obtain the best model fit (Hastie et al., 2009). This is performed recursively, until the activation of a stopping rule.

Assuming that the feature space has been partitioned into M regions, namely $R_1, \ldots, R_M$, and that the model's prediction at each region is c_m, the DTR model will have the fol-

lowing formulation:

$$y(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbf{1}\{x \in R_m\} \tag{4.21}$$

where **1** is the indicator function, returning 1 where the condition in brackets is true, and 0 in any other case. Following the same optimisation problem as with MLR, the best $\hat{c}_m$ can be obtained through the minimisation of the fit's LS, $\sum (y_i - f(x_i))^2$, obtaining as value the average of the observations lying in that region (Hastie et al., 2009):

$$\hat{c}_m = \text{ave}\,(y_i | x_i \in R_m) \tag{4.22}$$

Whilst the optimal c_m values can be easily computed, the same is not true for the region splitting. For this reason, a greedy algorithm is used recursively to find an optimal splitting, until the stopping rule is activated. This relates to the size of the tree and is a user-selectable parameter that relates to the data available and the complexity of the underlying problem. More specifically, user-selectable parameters are the maximum depth of the tree (reflecting the number of permitted splits), the minimum amount of samples required to split an internal node, the minimum amount of samples required to exist at each leaf , and the maximum amount of features considered when the splitting optimisation is performed.

4.6.1.5 K-Nearest Neighbours

Nearest Neighbours is one of the simplest non-parametric models. There, given a point x_q, the algorithm identifies the k nearest neighbours distance-wise (Russell & Norvig, 2010), with the parameter k being user-selectable.

Different algorithms exist for the computation of the nearest neighbours but Scikit-learn selects the most appropriate automatically, based on the input values. If a non-brute-force approach is used, an algorithm hyperparameter is leaf size that affects the speed and memory usage of the algorithm and depends on the underlying problem's nature.

In order to calculate the distance between x_q and any other point x_j, usually Minkowski

distance L^p is used

$$L^p(x_j, x_q) = \left(\sum_i |x_{j,i} - x_{q,i}|^p \right)^{1/p} \tag{4.23}$$

with $p = 1$ this corresponds to the Manhattan distance and with $p = 2$ to the Euclidean distance (Hu et al., 2016).

Additionally, the weighting function is user-selectable , as all k points can contribute equally ("uniform" weights) or the weight of each contributing point can be equal to the inverse of its distance from point x_q.

4.6.1.6 Support vector machines

SVMs in their simplest form constitute a two-class classifier in cases where the two classes are linearly separable. SVMs work by deriving the optimal hyperplane, i.e. the hyperplane that offers the widest possible margin between instances of the two classes. Their functionality can be extended by the introduction of a non-linear kernel, allowing them to learn non-linear mappings, i.e., classify between non-linearly separable classes (Theodoridis & Koutroumbas, 2009a). Depending on the properties of the selected kernel, SVMs can either be parametric or non-parametric models.

SVMs can also be built as regressors (Smola & Schölkopf, 2004). Support Vector Reggressors (SVRs) work in a similar way, this time trying to fit a hyperplane that accurately predicts the target values of training samples within a margin of tolerance ϵ. In the simpler case where a linear kernel is used, a SVR model will be of the form

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w} + w_0 \tag{4.24}$$

Model parameters w are obtained through the minimisation of the function

$$H(\mathbf{w}, w_0) = \sum_{i=1}^{N} V\left(y_i - f(x_i)\right) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{4.25}$$

where function V is defined as

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon & \text{otherwise} \end{cases} \tag{4.26}$$

and λ represents a regularisation term, similarly to, e.g. , LASSO models. This formulation of V allows errors of less than ϵ to be ignored (Hastie et al., 2009).

In the case of non-linear kernels, where the regression function is approximated in terms of a set of basis functions $\{h_m(x)\}$ where $m = 1, \ldots, M$, Equation 4.24 becomes of the form

$$f(\mathbf{x}) = \sum_{m=1}^{M} \mathbf{w}_m \, h_m(x) + w_0 \tag{4.27}$$

and accordingly, Equation 4.25 becomes

$$H(\mathbf{w}, w_0) = \sum_{i=1}^{N} V\left(y_i - f(x_i)\right) + \frac{\lambda}{2}\|\mathbf{w}_m\|^2 \tag{4.28}$$

An often-used non-linear kernel is the RBF kernel, formulated as

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{4.29}$$

In the ν-SVM implementation of the LIBSVM library (C. C. Chang & Lin, 2011) used in Scikit-learn, the penalty parameter of the error term is expressed by a parameter C, and the upper bound of the fraction of training errors is expressed by a parameter ν.

4.6.1.7 Shallow & deep neural networks

ANNs are an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron (Gurney, 1997). Nodes connect to each other based on inter-unit connection strengths called weights. ANNs can adopt different architectures that enhance their performance in machine learning

tasks, including classification and regression. Due to this, they are extremely versatile and can accurately model complex non-linear behaviours.

Tunable elements of a model can be grouped in two generalised categories, parameters and hyperparameters. Parameters are elements of the model that are directly learnt from the training data, whereas hyperparameters cannot be directly estimated from the training data as there is no analytical formula available to calculate an appropriate value (Kuhn & Johnson, 2013). Therefore, hyperparameters are specified manually, often following heuristics. The main hyperparameters that affect the performance of an ANN are the number of hidden (i.e., between input and output) layers, the number of nodes per layer, the activation function at each layer, the batch size, and in the case of deep ANNs the dropout rate.

The number of layers controls the depth of the ANN whilst the number of nodes per layer controls its width. Increasing the width of a layer increases its memorisation capabilities whereas increasing the depth of a network increases its capability of learning features at different levels of abstraction. In both cases, a more-is-better approach is not sensible as it would cause the model to overfit the training dataset.

Given an ANN regressor with an input layer $\mathbf{x}$, a hidden layer with M nodes Z_m, and an output layer consisting of a single node, each node is of the form (Hastie et al., 2009)

$$\begin{aligned} Z_m &= \sigma\left(\alpha_{0m} + a_m^T \mathbf{x}\right) \\ Y &= f(\mathbf{x}) = g(w_0 + w^T Z) \end{aligned} \tag{4.30}$$

where $Z = (Z_1, Z_2, \dots Z_M)$, $\sigma(\cdot)$ is the activation function and $g(\cdot)$ is the user-selectable output function. In the case of regression, the identity function, i.e. $g(x) = x$, is used as output function (Hastie et al., 2009). Currently sigmoid or ReLu (Rectified Linear Units) are the preferred activation function choices. The term sigmoid refers to a family of functions exhibiting a characteristic "S"-shaped curve. The formulation presented above can easily be extended for the case of multiple hidden layers by using the output

of each layer as input for the next, and so forth.

Dropout is applied in deep ANNs to reduce overfitting of models by randomly dropping nodes (and their connections) from the ANN during training (Srivastava et al., 2014). Dropout rate controls the probability of this random effect happening at each node. An ANN with and without dropout is depicted in Figure 4.8.

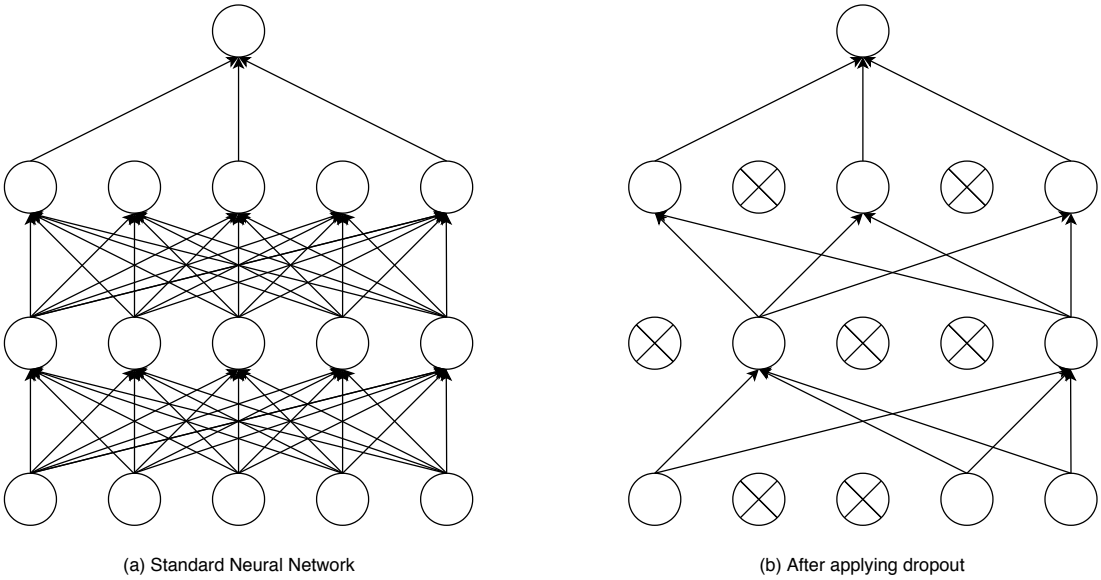

Figure 4.8: Example of a 2-layer ANN before (a) and after (b) the application of Dropout. Crossed units have been dropped and the dropout rate controls the probability of each unit dropping. Adapted from (Srivastava et al., 2014).

Batch size defines the number of training examples fed to the model before updating its parameters. A larger batch size will require a reduced computational cost but may converge to local minima instead of global (Goodfellow et al., 2016).

4.6.1.8 Random forest regressors

Random forests are based on the bagging meta-algorithm, where a number of decorrelated decision tree regressors are produced based on the available training set. Then, the output of the random forest regressor is calculated by averaging the results of individual decision trees.

4.6.1.9 Extra trees

Extra (extremely randomised) trees constitute a variation of Random Forest Regressors (RFRs) where the whole dataset is used at each instance (Breiman, 1998), and where the tree-splits are chosen completely at random.

4.6.1.10 AdaBoost

AdaBoost (adaptive Boosting) is a boosting meta-algorithm where a number of weak learners are combined into a weighted sum that represents the final output of the model.

4.6.2 Selection of optimal models

To test the regression model performance against the testing dataset, a number of metrics can be employed, each emphasising different model performance aspects. These will be analysed in the following subsections.

4.6.2.1 Explained variance

EV expresses the amount of variance that a model can capture from a given dataset. Having the true target output y, the estimated target output may be obtained as $\hat{y} = f(\mathbf{x})$, where $f(\cdot)$ refers to any derived model. Then, explained variance EV can be calculated as

$$EV(y, \hat{y}) = 1 - \frac{\sigma^2_{(y-\hat{y})}}{\sigma^2_y} \tag{4.31}$$

where σ_x refers to the standard deviation of parameter x. The best EV score is 1.0, obtained when $\sigma^2_{(y-\hat{y})} \to 0$, with lower values being worse.

4.6.2.2 Mean Absolute Error

Mean Absolute Error (MAE) corresponds to the expected value of the absolute (L^1 norm) error and can be calculated as

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.32}$$

where n refers to the number of samples in y, and y_i to the I-th sample of y.

A variant of MAE is Mean Absolute Percentage Error (MAPE), expressed in a percent form, as

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot 100\% \tag{4.33}$$

At first glance, MAPE seems to combine the benefits of MAE with an easier interpretation; in practice, a major drawback is that it becomes numerically unstable when there exists an i such that $y_i = 0$. However, there exists a ceiling of 100% error for under-estimated outputs, whereas no ceiling exists for overestimation. Due to this, underestimated forecasts are wrongly promoted, when comparing between models. For the above reasons, MAPE is not a considered model comparison metric in this study.

4.6.2.3 Mean Squared Error

Following the same formulation as above, the Mean Squared Error (MSE) can be calculated as

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.34}$$

MSE corresponds to the expected value of the quadratic error. Omitting the $\frac{1}{n}$ term, MSE becomes the L^2 loss function. Used as a cost function for optimisation purposes, both yield similar results.

Comparing to MAE, MSE puts a larger weight on major deviations between true and estimated targets. For the same reason, however, MAE remains more robust against outliers.

4.6.2.4 Mean Squared Logarithmic Error (MSLE)

The Mean Squared Logarithmic Error (MSLE) can be calculated as

$$MSLE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (\ln(1 + y_i) - \ln(1 + \hat{y}_i))^2 \tag{4.35}$$

MSLE tends to penalise more under-predictions rather than over-predictions. Furthermore, this loss function tends to under-penalise actual-estimated differences when both take large values; this can be of benefit when some observations momentarily take larger-than-usual values (e.g. full speed ahead at design draft).

4.6.2.5 Median Absolute Error

The Median Absolute Error (MedAE) can be calculated as

$$M_{ed}AE(y, \hat{y}) = \text{median}\left(|y_1 - \hat{y}_1|, \ldots, |y_i - \hat{y}_i|\right) \tag{4.36}$$

MedAE is especially robust to outliers due to only considering median performance.

4.6.2.6 Coefficient of Determination (R^2)

The coefficient of determination (R^2) can be computed as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.37}$$

where $\bar{y}$ is the mean value of y, i.e. $\bar{y} = \frac{1}{n}\sum_1^n y_i$.

R^2 provides a representation of the quality of future model output (predictions). The best R^2 score is 1, with lower values being worse. Furthermore, taking EV equation (Eq. 4.31), $\sigma^2_{y-\hat{y}}$ can be re-written into $\frac{1}{n}\sum_{i=1}^{n}\epsilon^2 - \bar{\epsilon}$, where $\epsilon = y - \hat{y}$. From that, we observe that when $\bar{\epsilon} \to 0$, EV's equation is transformed into Equation 4.37.

4.7 FOC prediction methodology

By applying the FOC modelling comparative methodology described in Section 4.6 Extra Trees Regressors (ETRs), RFRs, SVRs, and ANNs were identified as the models that yielded the best performance results. This is corroborated by the results of the relevant case study presented and discussed in Section 6.2. Out of those, only ANNs and SVRs offer the required generalisation capabilities required for weather routing. Com-

paring ANNs and SVRs, ANNs yielded better results when fed with ADLM-provided data. Therefore, in the methodology elaborated within this Section, a multi-layer (i.e., deep) ANN is developed for the purpose of FOC modelling. A thorough description of the ANN model derivation is presented in Section 4.6.1.7.

4.8 FOC-based performance monitoring methodology

Section 4.7 described an approach that, based on past voyages, learns how a vessel's FOC fluctuates depending on voyage particularities and ambient weather and predicts the FOC of future voyages. As a vessel's FOC is also affected by its hull and machinery condition, it is sensible to only feed a model with voyages that were made with the ship at a comparable condition. Therefore, when aiming to predict the FOC of voyages in the near future, recent data points are used for model training as those correspond to a relatively similar performance state.

However, if in the described approach, recent data points are substituted with data points corresponding to optimal performance, e.g. data that were gathered just post launching or following a hull cleaning and M/E overhaul, the resulting model will be able to predict the vessel's FOC assuming optimal performance. Comparing the observed FOC against the predicted, an estimation of the vessel's current performance can be estimated. This approach is visually depicted in Figure 4.9.

Depending on the end goal, a number of KPIs pertinent to vessel performance, efficiency and condition can be derived. Some of these are presented below, along with their mathematical formulation and application practicalities.

As the KPIs discussed below are based on the FOC modelling methodology developed in Section 4.7, a well-validated model is considered a prerequisite. Naturally, these KPIs are calculated at different points, across multiple routes and then averaged out over time-windows to create trendline plots of their deviation.

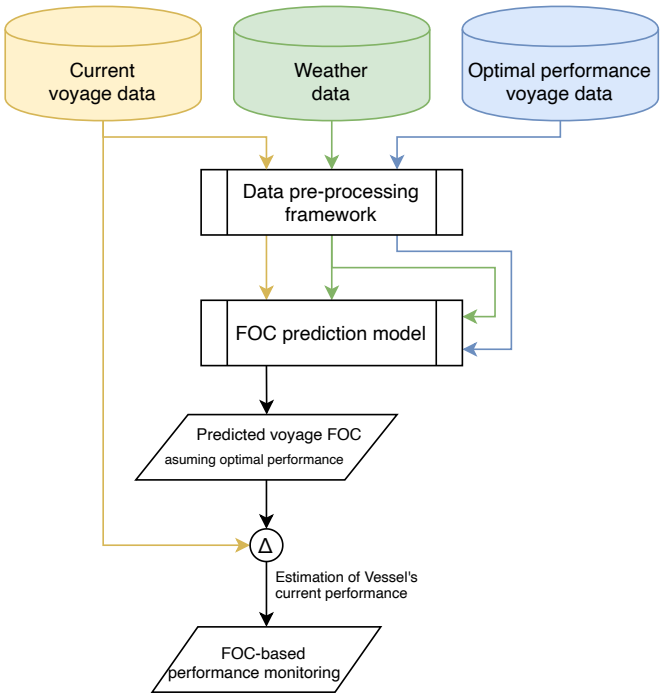

Figure 4.9: Visual representation of the proposed FOC-based performance monitoring methodology.

It should be noted that ideally, a separate model should be created for KPI purposes following the process described in Section 4.7 whilst utilising data of optimal performance. This is due to an inherent trade-off between the FOC prediction and Route Optimisation module requiring a FOC model that accurately predict a vessel's behaviour at its current state and the KPIs requiring a model that always reflects a vessel's optimal behaviour as a condition/performance baseline.

4.8.1 KPI I: Vessel Sailing Performance

The ratio between measured and expected distance travelled over any amount of fuel consumed can be used to quantify the sailing performance of a vessel.

Mathematically, this can be formulated as

$$KPI_I = \frac{d_{mf} - d_{ef}}{d_{ef}} \tag{4.38}$$

where d_{mf} refers to the measured distance over the specified fuel amount (e.g., 1 tonne) and d_{ef} to the model's estimation.

4.8.2 KPI II: Vessel Efficiency

For the purpose of tracking a vessel's sailing efficiency, the distance sailed using a specified amount of energy can be used as a suitable metric.

Mathematically, this can be formulated as

$$KPI_{II} = \frac{d_{me} - d_{ee}}{d_{ee}} \tag{4.39}$$

where d_{me} refers to the measured distance over the specified energy amount (e.g. 1 MWh) and d_{ee} to the model's estimation.

4.8.3 KPI III: Vessel Condition

To monitor the condition of the coupled hull and propeller system in the form of a KPI, the required power can be compared to the model's estimation. Therefore, this KPI can be mathematically formulated as

$$KPI_{III} = \frac{P_m - P_e}{P_e} \tag{4.40}$$

where P_m refers to the measured propulsion power for any sailing conditions and P_e to the model's estimation.

4.8.4 KPI IV: M/E Performance & Condition

In order to monitor the M/E condition and performance, a KPI based on the comparison between the measured and recorded Specific Fuel Oil Consumption (SFOC) values. This KPI can be mathematically formulated as

$$KPI_{IV} = \frac{SFOC_m - SFOC_e}{SFOC_e} \tag{4.41}$$

where $SFOC_m$ refers to the measured SFOC for any sailing conditions and $SFOC_e$ to the model's estimation.

4.9 Novel weather routing methodology

Sailing in adverse weather conditions can increase FOC and CO_2 emissions by over 50% (Prpić-Oršić et al., 2016). Furthermore, FOC constitutes approximately two-thirds of a vessel's voyage costs (Stopford, 2009). Therefore, the provision of optimal routing decision support, in terms of identifying a ship route that avoids adverse weather, can yield significant benefits in both financial and environmental terms.

Stemming from the above, the purpose of this section is to present a novel methodology for vessel weather routing based on historical ship performance and current weather conditions at a discretised grid of points. For this, a data-driven model that can predict main engine FOC is developed. Subsequently, a modified version of Dijkstra's algorithm that has been fitted with heuristics is applied recursively until an optimal route is obtained. This allows the creation of a vessel route that is derived based on the vessel's actual historical performance, increasing its robustness. The methodology developed in this section can be used to produce more accurate and robust weather routes, helping monitor and reduce vessels' FOC and CO_2 emissions. A flowchart that provides a high-level description of the proposed approach is presented in Figure 4.10.

Dijkstra's algorithm (Dijkstra, 1959) is an algorithm for identifying the shortest paths between nodes in a static graph. Dijkstra's algorithm in pseudocode form is presented in Algorithm 3.

In the case of vessel weather routing, the graph on which Dijkstra's algorithm is applied is dynamic as the cost of moving between nodes depends on the present weather conditions that vary as a function of time. For this reason, the following steps are followed to use Dijkstra's algorithm on a dynamic graph for weather routing:

Algorithm 3 Dijkstra's algorithm (Dijkstra, 1959)

Require: `graph`, `source`, `target`

```
1:  Q ← [ ]
2:  S ← [ ]
3:  for vertex v in graph do
4:      dist(v) ← ∞
5:      prev(v) ← NaN
6:      Add v to Q
7:  end for
8:  dist(source) ← 0
9:  while Q is not empty and u is not target do
10:     u ← vertex in Q that has minimum dist value
11:     remove u from Q
12:     for each neighbour v of u do
13:         if v in Q then
14:             alt ← dist(u) + length(u, v)
15:             if alt < dist(v) then
16:                 dist(v) ← alt
17:                 prev(v) ← u
18:             end if
19:         end if
20:     end for
21: end while
22: if prev(u) is not NaN or u is source then
23:     while u is not Nan do
24:         Add u at the beginning of S
25:         u ← prev(u)
26:     end while
27: end if
28: return S
```

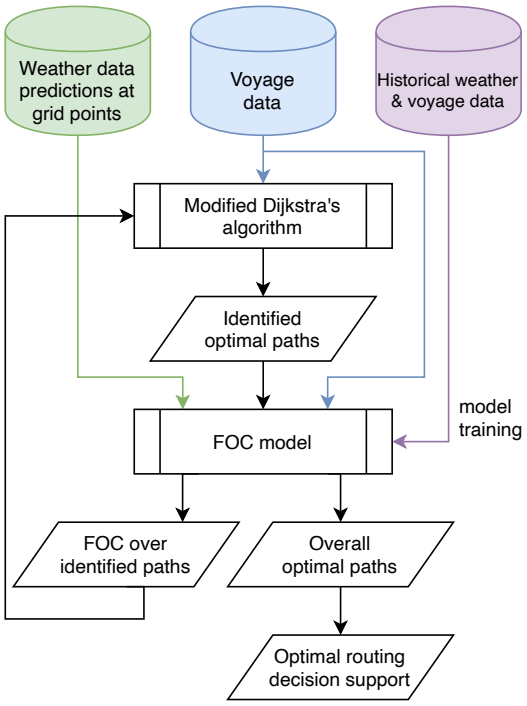

Figure 4.10: Visual representation of the proposed weather routing methodology.

1. Looping through all graph vertices v, apply Dijkstra's algorithm on the graph to identify the shortest path connecting source and target vertices through v_i. It should be noted that at this point, only the shortest distance is considered and not the vessel's FOC over the identified paths.

2. Calculate the vessel's FOC over the shortest paths, retain a user-selectable number of vertices v_{opt} that yield paths with lowest FOC.

3. Looping through v_{opt} and v, apply Dijkstra's algorithm on the graph to identify the shortest path connecting source and target vertices through v_{opt_i} and v_i.

4. Calculate the vessel's FOC over the shortest paths, retain a user-selectable number of vertice tuples v_{opt} that yield paths with lowest FOC.

5. Repeat steps 3–4 until no change in v_{opt} is observed.

This is also presented in pseudocode form in Algorithm 4.

Algorithm 4 Heuristic weather routing

Require: `M_FOC`, `graph`, `source`, `target`, n_i, n_l, n_o

```
all_paths ← [ ]
retained_internodes ← [ ]
for internode in graph do
    Apply Dijkstra's algorithm to identify shortest_path from source to target through internode
    if shortest_path is not in all_paths then
        Add shortest_path to all_paths
        Calculate its FOC using M_FOC
    end if
end for
retained_internodes ← n_i internodes yielding lowest FOC
for loop in n_l do
    for retained_internode in retained_internodes do
        for internode in graph do
            Apply Dijkstra's algorithm to identify shortest_path from source to target through retained_internode and internode
            if shortest_path is not in all_paths then
                Add shortest_path to all_paths
                Calculate its FOC using M_FOC
            end if
        end for
        Add n_i new internodes yielding lowest FOC to retained_internodes
    end for
end for
return n_o paths yielding lowest FOC
```

4.10 Chapter summary

A novel data-driven framework for the enhancement of a vessel's operational efficiency was presented in this chapter. As described throughout the chapter, the objective of this framework is four-pronged: a) provide machinery condition monitoring and anomaly detection; b) compare and identify optimal FOC modelling approaches; c) provide FOC-based performance monitoring at a vessel-level; and d) provide optimal routing decision support. The framework describes the modelling process end-to-end, from data collection, all the way to a novel heuristic that is proposed as an improvement over traditional Dijkstra's algorithm so it can used for optimal routing decision support purposes. Moreover, special focus is put on the derivation of the necessary models required for FOC-based performance monitoring of the vessel, and anomaly detection of individual machinery systems. In the following chapter, case studies are carried out to demonstrate the applicability, accuracy, and robustness of all individual methodological elements, and by extension, that of the overall framework.

Chapter 5

Case Studies' Description

This chapter presents the application of the data-driven framework elaborated in Chapter 4. Briefly recapping the novel operational efficiency enhancement framework presented in the previous Chapter, this comprises the following key elements: (a) a novel condition monitoring methodology; (b) a novel Fuel Oil Consumption (FOC) modelling comparative methodology; (c) a FOC-based performance monitoring methodology; and (d) an optimal routing decision support methodology. Due to the inherent inability of obtaining a single dataset that can be used for all these cases, these elements are presented and evaluated in separate case studies presented in the following sections.

5.1 Engine Condition Monitoring

The case studies elaborated in this section showcase applications based on the novel condition monitoring methodology presented in Section 4.5. The methodology is applied on one two-stroke main engine and a four-stroke engine used for electrical power generation. The aim of this case study is to evaluate the accuracy of the novel condition monitoring methodology in identifying incipient machinery anomalies.

These data are obtained from the noon reports of a 439 TEU reefer. The main particulars of this vessel are presented in Table 5.1. In particular, vessel noon reports of two

distinct time-periods are made available. One dataset contains both main engine and auxiliary generator data spanning a two-month period (Dataset AE1) whilst the other concerns a six-month period where only data of a single auxiliary engine (Dataset AE2) are made available. Hence, the first dataset contains 372 data points whilst the second contains 1095. Furthermore, there are minor differences in the sensor readings that each dataset contains due to changes in the noon-report format. Due to confidentiality reasons, both datasets are only presented in summarised form within this thesis.

Table 5.1: Main particulars of 439 TEU Reefer used in the Engine Condition Monitoring case study

Vessel particulars	
Year built	1993
Ship type	Reefer
TEU	439
Reefer points	70
Service speed	20.30 kn
Length overall	158.13 m
Beam	24.40 m
Depth moulded	15.37 m
Draft (summer)	10.00 m
Main engine	MAN B&W 7L60MC
Engine power	12,150 kW
Number of cylinders	7
Auxiliary power	3 × 1610 kW

A diagram of the main engine and the auxiliary engine systems corresponding to the AE1 dataset are shown in Figures 5.1 and 5.2 respectively, with the measurements considered highlighted in bold. Accordingly, the diagram of the auxiliary engine system contained in the AE2 dataset is depicted in Figure 5.3. It is noteworthy to mention that in this case, Air Cooler (AC) inlet and outlet temperature measurements are not available.

In all case studies considered herein, the diverse set of engine load conditions contained in the relevant datasets are used to train an One Class Classification (OCC) model capable of evaluating a number of D/Gen performance parameters to predict whether a given machinery snapshot corresponds to a "normal" or abnormal condition. A

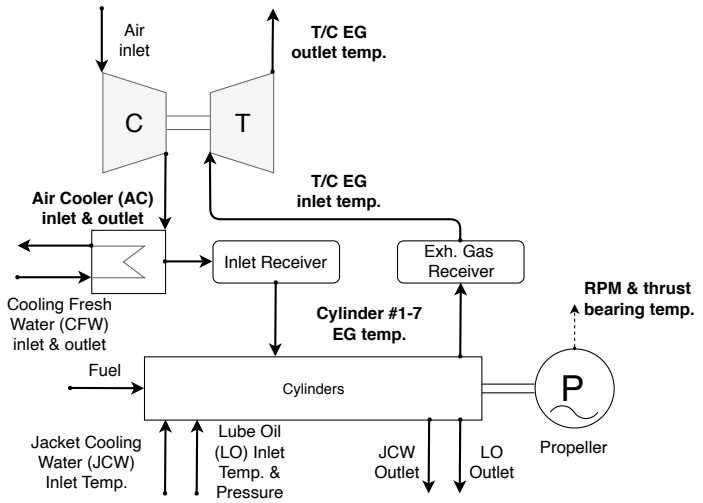

Figure 5.1: Diagram of the Main Engine (M/E) system used in the main engine condition monitoring case study. Parameter measurement locations are shown in bold.

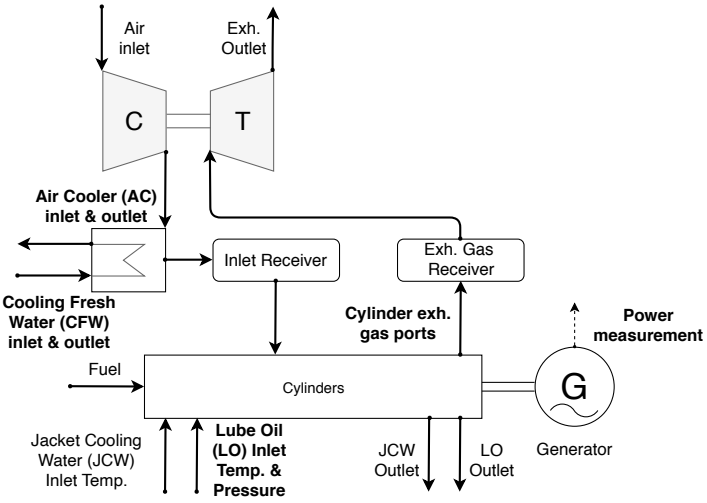

Figure 5.2: Diagram of the Diesel Generator set (D/Gen) system used in the auxiliary engine AE1 dataset condition monitoring case study. Parameter measurement locations are shown in bold.

sensitivity analysis is then performed on the model, evaluating model behaviour under a number of simulated scenarios to ensure its robustness.

Based on the above, three case studies of interest can be identified relating to the novel condition monitoring methodology proposed within this thesis. These are: (a) main engine condition monitoring; (b) auxiliary engine condition monitoring based on dataset

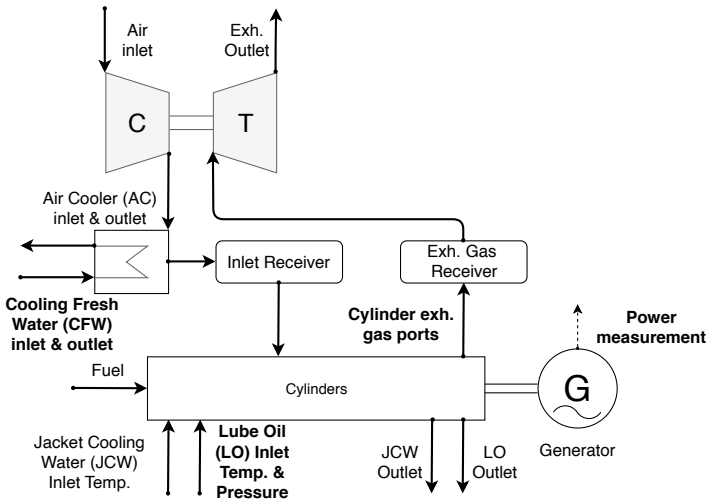

Figure 5.3: Diagram of the D/Gen system used in the auxiliary engine AE1 dataset condition monitoring case study. Parameter measurement locations are shown in bold.

AE1; and (c) auxiliary engine condition monitoring based on dataset AE2.

5.1.1 Verification

Model verification is a challenging task, especially in cases where modelling is performed at system-level, as it is generally extremely difficult or outright impossible to model the entire possible input domain and therefore, a simulation model can only be an approximation to the actual system regardless of effort spent on the model development (Law, 2006). Even when a model is considered to be validated against an observable system, in cases where the model is used for alternative configurations, the underlying assumptions may no longer be valid (Dinwoodie, 2014).

As accurate raw data are not available, to verify the proposed methodology, sensitivity analysis is employed, considering multiple sensitivity case scenarios. The sensitivity analysis process examines the flexibility of the underlying methodology in identifying dissimilarities between nominal and unclassified observations and consequently increase the probability of abnormal condition.

5.1.2 Main engine dataset

The variables considered as input data for this main engine case study are presented in Table 5.2. In this case, the engine shaft revolutions are recorded, along with the thrust bearing Lube Oil (L.O.) outlet temperature, the AC inlet and outlet temperature, the input and output Turbocharger (T/C) air temperature, and the Exhaust Gas (EG) temperature at each cylinder. In this case, noon reports containing 372 observations are available, corresponding to 62 days of recording, with a 6 points per day sample rate.

Table 5.2: Noon-report measurements considered as input for the main engine condition monitoring case study.

Component	description	units
Misc.	shaft revolutions	RPM
Lub. Oil	Thrust bearing L.O. outlet temp.	°C
Receiver scavenge air	AC inlet temp.	°C
	AC outlet temp.	°C
EG	T/C inlet temp.	°C
	T/C outlet temp.	°C
	Cyl #1-7 EG temp.	°C

As part of model training, hyperparameter optimisation is performed through the grid search optimisation approach. Furthermore, in order to verify algorithm performance, sensitivity analysis is performed across key variables.

5.1.2.1 Design of synthetic fault data

In order to evaluate model diagnostic performance, the testing dataset is augmented with simulated faults (i.e. anomalies) in the form of a sensitivity analysis. To achieve that, for each selected attribute in the testing dataset, the testing dataset is replicated and then the values of that attribute are linearly altered in order to reach the alarm or shutdown thresholds set by the Original Equipment Manufacturer (OEM). The two testing datasets are then concatenated and fed to the trained condition monitoring model as input with its output being observed. Expected model behaviour is to return a positive distance from the hyperplane (i.e. normal behaviour) for points belonging to

the "normal" dataset and then have that distance decrease, cross zero, and eventually take negative values (i.e. abnormal behaviour) as the manipulated attribute exceeds the normal range.

Depending on the selected attribute, a different manipulation range is implemented, to reflect realistic failure conditions. The range selected per parameter are displayed below, in Table 5.3.

Table 5.3: Deviation-from-norm ranges considered in the production of the synthetic faults dataset for model evaluation in the main engine condition monitoring case study.

Attribute	Range
Max cyl EG temp.	+0°C to +25°C
Mean cyl EG temp.	+0°C to +25°C
Mean cyl EG standard deviation (σ)	+0°C to +7°C
Shaft revolutions	-0 RPM to -10 RPM
T/C inlet/outlet temperature Δ	-0°C to -40°C
Thrust bearing L.O. outlet temp.	+0°C to +7°C

5.1.2.2 Model training

The methodology proposed in Section 4.5 is applied in the model training phase of this case study. Hence, sensor data corresponding to nominal operating conditions are used for model training. These data are pre-processed by applying the steps discussed in Section 4.3, namely: a) rejection of transients, and recording anomalies; b) feature engineering (International Organization for Standardization (ISO) correction of necessary measurements, and derivation of temperature differentials in the case of the T/C); and c) data standardisation.

In order to evaluate different hyperparameter optimisation approaches, grid search is applied in this case, evaluating γ and ν in the range $(0, 1]$, and $[2^{-15}, 2^{3}]$ respectively, following the recommendation of Hsu et al. (2010). A 100 $\times$ 100 grid is selected, leading to the evaluation of 10,000 hyperparameter combinations in total. This value is chosen as 10,000 draws can provide a set within 0.1% of optimal with 99.99% (i.e. $1 - (1 - 0.001)^{10000} \times 100\%$) confidence, irrespective of grid size. Opting for 3000 combinations, as is the case for the following condition monitoring case studies, would

have yielded similar model training results.

5.1.3 Auxiliary engine dataset AE1

The variables considered as input data for this case study are presented in Figure 5.4. Specifically, minimum and maximum EG temperature measurements are provided, inlet and outlet temperature of the scavenge air receiver, fresh water cooler inlet temperature, lubricating oil inlet temperature and pressure as well as power output. As previously mentioned, noon reports containing 372 observations are available, corresponding to two months of recording, at a 6 points per day sample rate.

Table 5.4: Noon-report measurements considered as input for the auxiliary engine AE1 dataset condition monitoring case study.

Component	description	units
Misc.	Power output	kW
Lub. Oil	L.O. inlet temperature	°C
	L.O. inlet pressure (manometric)	bar
Fresh Water Cool.	Cooling Fresh Water (CFW) inlet temperature	°C
Cylinder EG	Max temperature	°C
	Min temperature	°C

5.1.3.1 Design of synthetic fault data

Similarly to the previous condition monitoring case study presented above, synthetic fault data are created in this case study to evaluate the model's performance. The same process as above is followed, i.e. the testing dataset is replicated, and augmented by selecting one attribute each time and linearly manipulating it so that it exceeds the normal operation limits stipulated by the OEM. In this case, the ranges applied to each variable are presented in Table 5.5. Parameter range values are selected aiming to transcend OEM limits in order to test the model's anomaly detection capabilities.

5.1.3.2 Model training

As discussed in the methodology Section 4.5 to achieve the aim of this methodology, i.e. detect anomalous machinery operation, an OCC Support Vector Machine (SVM) model is trained using sensor data corresponding to a nominal engine condition. These

Table 5.5: Deviation-from-norm ranges considered in the production of the synthetic faults dataset for model evaluation in the auxiliary engine AE1 dataset condition monitoring case study.

Attribute	Range
L.O. inlet temp.	100% – 120%
L.O. inlet press.	75% – 100%
CFW inlet temp	100% – 130%
EG temp max	100% – 160%

data are preprocessed following the methodology outlined in Section 4.3, therefore the following steps are implemented: a) engine transients are rejected; b) recording anomalies are rejected; c) feature engineering is applied, namely ISO correction of pertinent variables; and d) data standardisation is performed.

The parameters considered as input to the model have been presented in Table 5.4. Moreover, following the formulation discussed in Section 4.5 and Subsection 4.5.1, hyperparameters γ and ν need to be optimised in the range $(0, 1]$, and $(0, \infty)$ respectively. However, Hsu et al. (2010) propose that ν is optimised in the $[2^{-15}, 2^{3}]$ range, significantly simplifying the optimisation problem. Random search is applied for hyperparameter optimisation, considering 3000 (ν, γ) sets. 3000 iterations are selected, as 3000 draws can provide a set within 0.5% of optimal with 99.99% (i.e. $1 - (1 - 0.005)^{3000} \times 100\%$) confidence, irrespective of grid size.

5.1.4 Auxiliary engine dataset AE2

The variables considered for this two-month auxiliary engine case study are depicted in tabular form in Table 5.6. This case study considers the following variables: minimum and maximum EG temperature measurements are provided, inlet and outlet temperature of the scavenge air receiver, fresh water cooler inlet temperature, lubricating oil inlet temperature and pressure, AC inlet and outlet temperature, as well as power output. In this case, noon reports containing 1095 observations are available, corresponding to 182.5 days of recording, with a 6 points per day sample rate.

Following the paradigm of the case study elaborated in Section 5.1.2, grid search is

Table 5.6: Noon-report measurements considered as input for the auxiliary engine AE2 dataset condition monitoring case study.

Component	description	units
Misc.	Power output	kW
Lub. Oil	L.O. inlet temperature	°C
	L.O. inlet pressure (manometric)	bar
Fresh Water Cool.	CFW inlet temperature	°C
Cylinder EG	Max temperature	°C
	Min temperature	°C
Receiver scavenge air	AC inlet temp.	°C
	AC outlet temp.	°C

applied for hyperparameter optimisation. Additionally, the algorithm performance and robustness is evaluated by applying sensitivity analysis across key variables.

5.1.4.1 Design of synthetic fault data

In this case study, the maximum EG temperature observed across all cylinders is linearly increased to simulate an operating point that exhibits a fault. As in previous case studies, the original dataset is replicated and augmented by this simulated parameter. The sensitivity analysis of this parameter is performed following the ranges described in Table 5.7.

Table 5.7: Deviation-from-norm ranges considered in the production of the synthetic faults dataset for model evaluation in the auxiliary engine AE2 dataset condition monitoring case study.

Attribute	Range
Max cyl EG temp.	+0°C to +30°C
Thrust bearing L.O. outlet temp.	+0°C to +7°C

5.1.4.2 Model training

The methodology proposed in Section 4.5 is applied in the model training phase of this case study, in line with the process performed in the previous Condition Monitoring (CM) case studies. Sensor data corresponding to nominal operating conditions are used for model training. These data are pre-processed by applying the steps discussed in Section 4.3, i.e.: a) rejection of transients; b) rejection of recording anomalies; c) feature

engineering (i.e., ISO correction of necessary measurements); and d) data standardisation.

Grid search is applied for the identification of optimal hyperparameters, evaluating γ and ν in the range $(0, 1]$, and $[2^{-15}, 2^3]$ respectively (Hsu et al., 2010). A 100×100 grid is considered, leading to the evaluation of 10,000 hyperparameter combinations in total.

5.2 FOC modelling comparison

The two case studies contained within this section prove the applicability of the novel FOC modelling comparative methodology described in Section 4.6, i.e. the performance comparison of data-driven regression models in the estimation of a vessel's FOC under different circumstances. The aim of this case study is to identify optimal FOC modelling approaches when different data sources exist, namely noon reports and Automated Data Logging & Monitoring (ADLM) systems.

The first case study concerns data data acquired from a reefer ship (Vessel 1 – V1) whilst the second is based on data from a Newcastlemax Bulk carrier (Vessel 2 – V2). While Vessel 2 is equipped with an ADLM system, in the case of Vessel 1, noon-report data are used. Similar input parameters are considered for both vessels (Table 5.8), albeit at different sampling rates. The main particulars of both vessels are presented in Table 5.9.

To increase the transparency of the included case studies, the exact roadmap followed to obtain the results is as follows

1. Dataset is loaded.
2. Unneeded features are discarded.
3. Engine transients are identified and discarded.
4. Observations containing Not a Number (NaN) elements are identified and dis-

Table 5.8: Dataset measurements considered in the FOC modelling comparison case study. V1 refers to the noon-report dataset and V2 refers to the ADLM dataset.

#	Name	V1 Units	V2 Units
1	Vessel speed	knots	knots
2	Engine speed	rev/min	rev/min
3	Sea current [i]	knots	knots
4	Wind speed	Beauford scale	m/s
5	Wind direction [i]	12 direction bins	degrees
6	Daily M/E FOC	t/day	t/day
7	Daily distance run	nm	nm
8	Sea state	Douglas sea scale	m[ii]
9	Sea direction [i]	12 direction bins	degrees
10	Slip	%	%
11	Draft fore	m	m
12	Draft aft	m	m

[i] relative to vessel [ii] wave height

carded.

5. Extract additional features (e.g. FOC, current, distance run, slip).

6. Discard points where slip ≤ 0, caused by round-off errors.

7. Split dataset into training and test set.

8. Scale training set and utilise same scaling parameters for test set.

9. Populate list of potential models

10. For every model in list of models implement k-folding cross-validation and

    (a) train using default Scikit-learn hyperparameters.

    (b) train using hyperparameter optimisation.

        i. identify search space for each hyperparameter.

        ii. run random search over the search space and evaluate results.

Table 5.9: Main particulars of two vessels considered in the FOC modelling comparison case study. V1 refers to the noon-report dataset and V2 refers to the ADLM dataset.

Vessel particulars	V1	V2
Year built	2016	2016
Ship type	Reefer	Bulk Carrier
TEU	1100	-
Length overall	147.00 m	299.88 m
Beam	24.00 m	50.00 m
DWT	13,000 tons	208,000 tons
Main engine	MAN 6S40ME-B	MAN 6G70ME-C
Engine power	6900 kW	17,500 kW
Number of cylinders	6	6
Data source	Noon reports	ADLM

(c) Using optimal hyperparameters, train a model using the whole training set (no validation).

(d) Evaluate model results on test set and compute performance metrics.

11. Evaluate all models based on metrics and reach overall conclusions.

Both vessel datasets are filtered to only include observations with the M/E speed being above the OEM lowest continuous running limit. This filtering is applied in order to only take into account the data points that correspond to relatively steady state conditions, without significant transient instances, e.g. manoeuvring.

The datasets are split 80–20% for training and testing data. While this value is different compared to the 70–30% percent split performed in the case studies of Section 5.1.3, it remains in line with data science best practices, when applied to relatively small datasets.

Datasets are then normalised following the steps elaborated in Section 4.3 in order to be used as training input for all relevant models. Each model are then trained using the default Scikit-learn (Pedregosa et al., 2011) hyperparameters. Additionally, random search over hyperparameters pertinent to each model is performed to identify optimal values.

5.2.1 Model hyperparameter optimisation & selection of optimal model

Considering the benefits provided by random search, a random search optimisation loop is set-up for all models. 1000 iterations are used for the hyperparameter optimisation of all models.

In order to reasonably ensure that selected hyperparameter values are actually close to optimal and not merely overfitting the model, cross-validation is implemented in the form of k-folding. Using the same technique for all models, allows the identification of a model that performs best while at the same time ensuring good generalisation capabilities.

In order to identify optimal models and hyperparameters, the coefficient of determination (R^2) (Glantz & Slinker, 2000) is evaluated for each model produced at each fold. For the evaluation of models post-training, a number of metrics are calculated, as discussed in Section 4.6.2. The hyperparameters that are considered for each model and their range of values are presented in Table 5.10.

Moreover, model training is repeated 10 times and timed, so that an estimation of the training time of each model can be obtained. The times recorded for each model and dataset are averaged for the two datasets in order to get one common, consistent result.

The results of this case study will allow the identification of optimal models for the prediction of vessel FOC depending on the data source. Moreover, model accuracy differences based on data source will be quantified.

5.3 Vessel FOC prediction & weather routing

This case study concerns a 160,000 tonne DWT crude oil tanker sailing between the Gulf of Guinea and the Marseille anchorage, and can be extended as-is to any other ship type and voyage. This above route is selected as it represents a typical voyage

Table 5.10: Hyperparameters and relevant range considered for each model in the FOC modelling comparison case study.

Model	**Hyperparameters tuned**
Least Absolute Shrinkage and Selection Operator (LASSO)	$\lambda \in \left[10^{-7}, 10^{5}\right]$
Ridge Regression (RR)	$\lambda \in \left[10^{-7}, 10^{5}\right]$
Elastic net	$\alpha \in \left[10^{-7}, 10^{5}\right]$, $\lambda 1_{ratio} \in [0, 1]$
Decision Tree Regressor (DTR)	`max_tree_depth` $\in \left[10^{-7}, 10^{5}\right]$, `min_samples_split` $\in [2, 20]$, `min_samples_leaf` $\in [3, 20]$, `max_features` $\in [3, 10]$
Random Forest Regressor (RFR)	`n_estimators` $\in [1, 200]$, `min_samples_split` $\in [2, 20]$
K-Nearest Neighbours (KNN)	`n_neighbours` $\in [1, 50]$, `weights` $\in [\text{uniform}, \text{distance}]$, `leaf_size` $\in [2, 100]$
Support Vector Reggressor (SVR)	$\gamma \in 2^{[-15,3]}$, $C \in 2^{[-5,5]}$, $\nu \in \left[10^{-4}, 1\right]$
Extra Trees Regressor (ETR)	`n_estimators` $\in [1, 100]$, `max_features` $\in \left[10^{-5}, 1\right]$, `min_samples_split` $\in \left[10^{-5}, 1\right]$, `max_samples_leaf` $\in \left[10^{-5}, 0.5\right]$
Artificial Neural Network (ANN)	`activation` $\in [\text{relu}, \text{tanh}]$, `alpha` $\in \left[10^{-8}, 10^{-1}\right]$, `hidden_layer_sizes` $\in [1, 50]$ or $[1, 50; 1, 50]$

of the aforementioned vessel. The aim of this case study is to (a) ascertain that the FOC of this vessel can be accurately predicted under varying sailing and operational conditions; and (b) demonstrate that the novel weather routing methodology proposed in Section 4.9 can accurately identify optimal sailing routes for the investigated voyage. The principal particulars of the vessel are displayed in Table 5.11.

Data pertaining to 10 ship voyages are provided for FOC model training and testing. In total, 2713 data points are obtained after merging all provided information. The parameters obtained are presented in Table 5.12.

The ocean analysis & forecast data used for weather routing contains measurements 7 – 14 of Table 5.12 at 368 ocean locations (Figure 5.4) corresponding to a graph that encompasses the source and target navigation points at a resolution of 1° in both latitude and longitude at a time frequency of 3 hours. Intermediate points are interpolated

Table 5.11: Main particulars of vessel considered in the vessel FOC prediction and weather routing case study.

Vessel particulars	
Year built	2016
Ship type	Crude oil carrier
Length overall	274 m
Beam	48.00 m
DWT	159,000 tons
Main engine	MAN 6G70ME-C
Engine power	16,600 kW
Number of cylinders	6
Data source	ADLM

in space and time.

The overall architecture of the developed ANN is depicted in Figure 5.5 and the search space of the model's hyperparameters in Table 5.13. This model is selected as it is one of the best performing models for this application, as identified in Section 6.2. Tree-based models (e.g. ETRs, RFRs) are not considered due to their poor extrapolation abilities. Moreover, due to the ample number of data points available in this case study, more complex ANN architectures are investigated. Indicatively, the existence of hidden layers 3 and 4 depends on the selected value of network depth. Accordingly, the width, dropout rate, and activation function are considered individually for each layer.

Due to the high computational cost of testing every possible combination of hyper-parameters to identify the optimal, an optimisation algorithm is used. In this case, a Tree-structured Parzen Estimators (TPEs) approach is applied (Bergstra et al., 2011) due to the increased computational cost of model training.

Any number of available measurements x_i are used as model input and the model's output is the FOC for that route. It should be noted at this point that the same input measurements used at training must also be available at execution time for the model to work as expected.

Following FOC model training, the weather routing methodology is applied for two

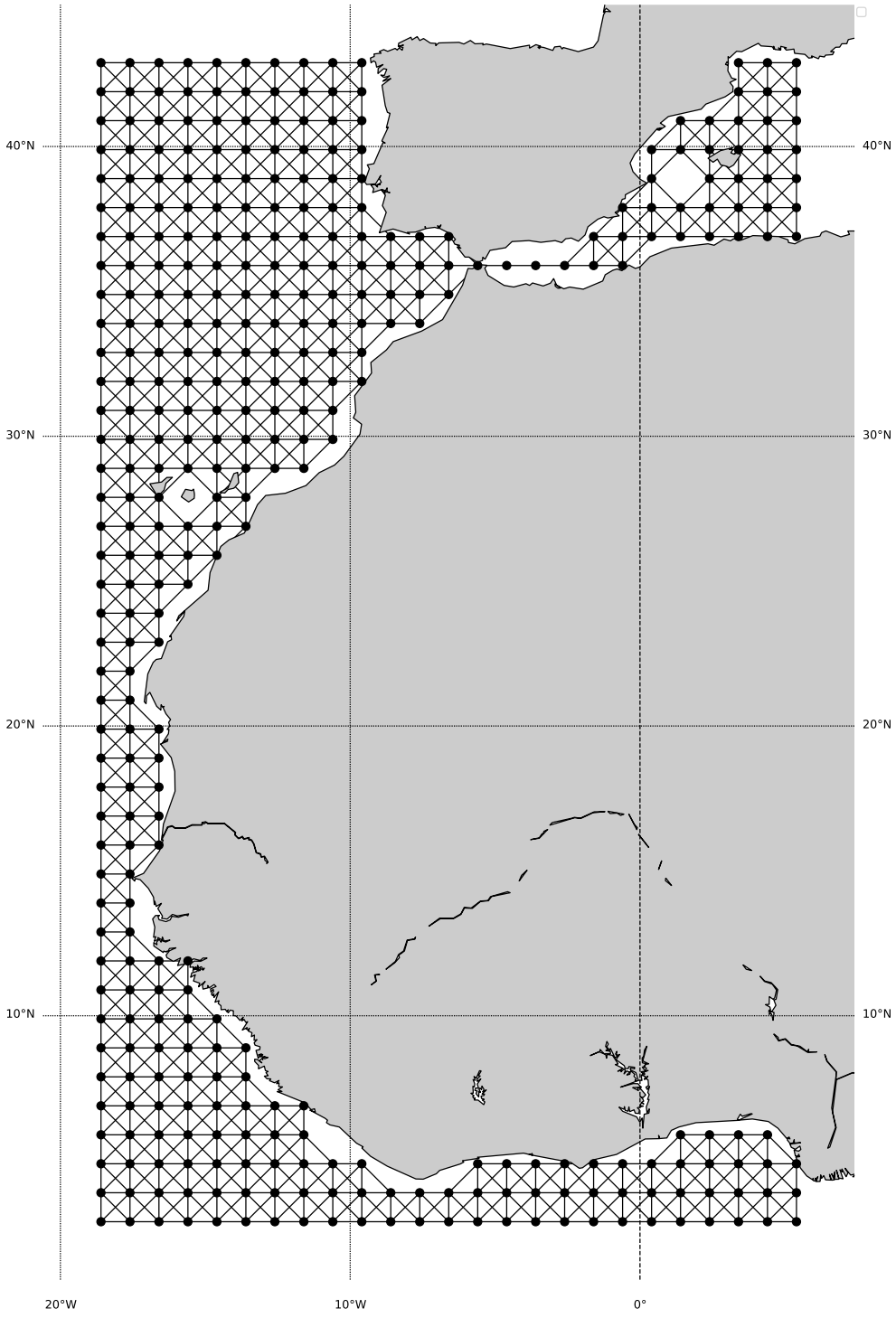

Figure 5.4: Visual representation of graph points where weather information is available in the vessel FOC prediction and weather routing case study.

Table 5.12: Overview of vessel sailing dataset parameters used for model training in the vessel FOC prediction and weather routing case study.

Name	Units
Draft fore	m
Draft aft	m
Engine speed	rev/min
Engine FOC	kg/hr
Engine power	kW
Vessel over ground speed	knots
Eastward current	knots
Northward current	knots
Primary swell wave significant height	m
Secondary swell wave significant height	m
Primary swell wave direction	degrees
Secondary swell wave direction	degrees
Wave significant height	m
Wave direction	degrees

Table 5.13: ANN hyperparameter search space for the vessel FOC prediction and weather routing case study. 2,3 or 4 hidden layers of varying width were considered, along with different dropout rates, activation functions, and batch sizes.

Hyperparameter	Search space
Network depth	{2, 3, 4}
Layer width	{8, 16, 32, 64, 128, 256, 512}
Dropout rate	[0, 1)
Activation function	{ReLU, Sigmoid}
Batch size	{32, 64, 128}

case studies where the vessel is sailing at two speeds, 11 knots and 14.5 knots. The aforementioned vessel speeds refer to two typical scenarios of vessel speed whilst fully ladden, and in ballast condition.

5.4 FOC-based performance monitoring

The case study elaborated within this Section stems from the FOC-based performance monitoring methodology described in Section 4.8. This methodology aims to provide vessel performance monitoring using readily available data, including noon reports. Specifically, a data-driven FOC predictive model is built based on data obtained just

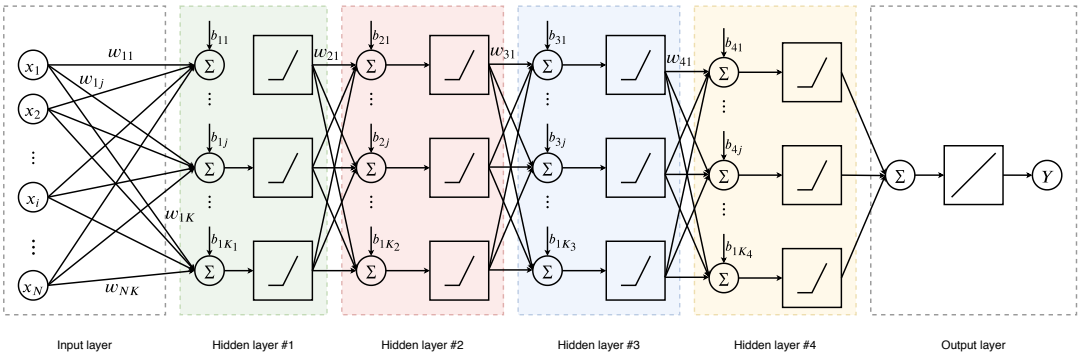

Figure 5.5: Visual representation of the developed ANN architecture for the vessel FOC prediction and weather routing case study. Coloured boxes represent different hidden layers, boxes with straight or obtuse-angled lines represent activation functions and a Σ inscribed in a circle represents the summation function.

after the launching of the investigated vessel and the model results are compared to the vessel's actual performance in terms of FOC.

The vessel considered in this case study is a 1100 TEU reefer vessel, sister vessel of the one considered in Section 5.2. Its main particulars are presented in Table 5.14.

Table 5.14: Main particulars of vessel considered in the FOC-based performance monitoring case study.

Vessel particulars	
Year built	2016
Ship type	Reefer
TEU	1100
Length overall	147.00 m
Beam	24.00 m
DWT	13,000 tons
Main engine	MAN 6S40ME-B
Engine power	6900 kW
Number of cylinders	6
Data source	Noon reports

A time period of 27 months is considered at a daily frequency, totalling 823 data points.

For regression, two models are applied: a Linear Regression (LR) model as a baseline, and a SVR model including the optimisation of its hyperparameters. These models are

selected due to their overall simplicity whilst providing good results, as observed in the case study of Section 5.2. Moreover, as the datasets used in this case study need to be short in order to reflect the examined vessel's condition, models that can perform well using a limited amount of training data points were required.

Hyperparameter optimisation is performed through cross validation, with $k = 10$ folds and 1000 iterations per fold. Similarly to previous case studies, the guidelines of Hsu et al. (2010) are applied, and random search is applied, considering C values in the range $[2^{-5}, 2^{5}]$, γ values in the range $[2^{-15}, 2^{3}]$, and ν values in the range $[0, 1]$.

Both models are trained using data from the first six months following the vessel's launch and evaluated at three distinct time points: seven months post-launch, i.e. the time just following the window used for model training; eighteen months post-launch; and twenty-five months post launch. In all time points, a full month is evaluated by comparing the model's output with the actual vessel FOC during the same period.

5.5 Chapter summary

This chapter presented an overview of the case studies considered for the evaluation and validation of the operational efficiency enhancement framework proposed in Chapter 4. This framework comprises four key methodologies, namely: (a) the novel condition monitoring methodology; (b) the novel FOC modelling comparative methodology; (c) the FOC-based performance monitoring methodology; and (d) the optimal routing decision support methodology. The condition monitoring methodology aims to provide anomaly detection using machinery process data. Accordingly, the FOC modelling comparative methodology, aims to identify optimal FOC modelling architectures depending on the available data sources. The weather routing methodology aims to provide data-driven optimal routing decision support for in a variety of operating conditions. the FOC-based performance monitoring methodology aims to provide vessel performance monitoring using readily available data, including noon reports. The case studies elaborated in this section reflect these four case studies and are designed to

exhibit the capabilities of the underlying framework in realistic operating conditions. Following the above, the next Chapter presents and discusses the results based on the case studies of this Chapter.

Chapter 6

Case Studies' Results & Discussion

This chapter presents and discusses the results obtained from the case studies elaborated in Chapter 5. Initially, the results of the main and auxiliary engine condition monitoring case study detailed in Section 5.1 are presented in Section 6.1. Following that, the results obtained from the Fuel Oil Consumption (FOC) modelling comparison case study of Section 5.2 are demonstrated in Section 6.2. Subsequently, based on the evaluation of the results of the FOC modelling comparison case study, the results of the vessel FOC prediction & weather routing case study of 5.3 are presented in Section 6.3. Furthermore, the results of the FOC-based performance monitoring case study originally presented in Section 5.4 are presented in Section 6.4. Finally, an overall discussion on the results presented and individually discussed in this chapter is provided in Section 6.5, followed by the Chapter summary in Section 6.6.

6.1 Engine Condition Monitoring

This section presents and discusses the results obtained for the engine condition monitoring case study originally described in Section 5.1. Three subsections will follow, each

elaborating on one of the three cases considered in this case study, i.e. (a) main engine condition monitoring; (b) auxiliary engine condition monitoring based on dataset AE1; and (c) auxiliary engine condition monitoring based on dataset AE2.

When model training is performed using a training dataset that only includes a limited subset of vessel's operating conditions, model output given a specific input can be seen as the probability of this input being identified as abnormal. However, if model training is performed using a dataset that covers most vessel's operating conditions and has been identified by the OEM or operator as satisfactory, model output can be seen as a metric of system health. In parallel, assuming a dataset that covers satisfactorily most of the vessel's operating conditions, the sensitivity analysis performed can be seen as an indicative tolerance between current conditions and a soft bound of acceptable performance. In this sense, model provides an estimation of how much a parameter deviation indicates performance degradation. The probability warning/safety limit, beyond which maintenance actions are required, is vessel and system specific. This limit is identified depending on stakeholders' interpretation and requirements.

Accordingly, a sufficiently large and diverse testing dataset confirms the accuracy of the trained model. Furthermore, such a dataset confirms that a good fit has been achieved without the occurrence of over-fitting.

6.1.1 Main engine dataset

Discarding 131 Not a Number (NaN) values and outliers following the process elaborated in Section 4.3.2, 243 data points are available. Furthermore, in this case a 70%–10%–20% three-way split between training, validation, and testing is performed. Validation refers to the part of the dataset used to evaluate the performance of different hyperparameters while a model is trained using the training dataset. This happens in order to ensure that the testing dataset remains unseen until the final evaluation of the model so that its generalisation capabilities can be evaluated. This process produced a training set containing 170 observations, while the validation and testing datasets contained 24 and 49 data points respectively.

The descriptive statistics of the training dataset used in this case study is shown in Table 6.1, where the mean, minimum and maximum values of each feature are presented, along with the standard deviation and the values at different quantiles. These values 1 constitute a factual representation of the dataset used and are presented to provide key details of the dataset, as the whole dataset cannot be presented due to confidentiality reasons. Statistical calculations such as those contained in this Table only have descriptive power and cannot be used to make predictions, contrary to Machine Learning (ML) models developed and presented in this case study.

Table 6.1: Descriptive statistics of the training part of the dataset used in the main engine condition monitoring case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Engine Speed (RPM)	Air Cooler (AC) inlet temp. (°C)	AC outlet temp. (°C)	Cyl #1 Exhaust Gas (EG) temp (°C)	Cyl #2 EG temp (°C)	Cyl #3 EG temp (°C)	Cyl #4 EG temp (°C)
count	243	243	243	243	243	243	243
mean	30.52	44.56	140.58	44.52	360.21	361.28	362.00
std	17.26	3.02	11.24	3.02	8.94	9.08	8.77
min	1.16	40.00	124.00	40.00	335.00	330.00	340.00
25%	18.12	42.00	134.00	42.00	355.00	355.00	355.00
50%	30.91	44.00	137.00	44.00	360.00	360.00	360.00
75%	44.37	45.00	142.00	45.00	365.00	370.00	370.00
max	61.83	56.00	167.00	56.00	380.00	380.00	380.00
	Cyl #5 EG temp (°C)	Cyl #6 EG temp (°C)	Cyl #7 EG temp (°C)	Turbocharger (T/C) inlet temp (°C)	T/C outlet temp (°C)	Thrust bearing Lube Oil (L.O.) temp (°C)	
count	243	243	243	243	243	243	
mean	360.68	361.50	363.70	354.94	404.52	2.26	
std	8.80	9.53	7.48	8.26	9.68	0.07	
min	345.00	338.00	350.00	340.00	385.00	2.00	
25%	355.00	355.00	360.00	350.00	395.00	2.20	
50%	360.00	360.00	360.00	355.00	400.00	2.30	
75%	366.25	365.00	370.00	360.00	410.00	2.30	
max	385.00	380.00	380.00	380.00	430.00	2.40	

In Figure 6.1, the results of grid search employed to determine the optimal values of hyperparameters ν and γ is shown. Lighter colours correspond to better performance and conversely. Values towards the manually-imposed lower bounds of both hyperparameters seem to return better results, with optimal performance being obtained for $\nu = 0.250$ and $\gamma = 0.050$. This is justified by the relative lack of complexity in training data as they all correspond to nominal condition. Meanwhile, this highlights the importance of setting suitable upper and lower limits during grid search, taking into consideration the nature of training data.

In Figures 6.2 and 6.3 model performance is presented in cases where the maximum or mean exhaust gas temperature is increased. In both cases, an increase of over +15°C returns a probability of abnormality of about 60%, with a probability of 100%

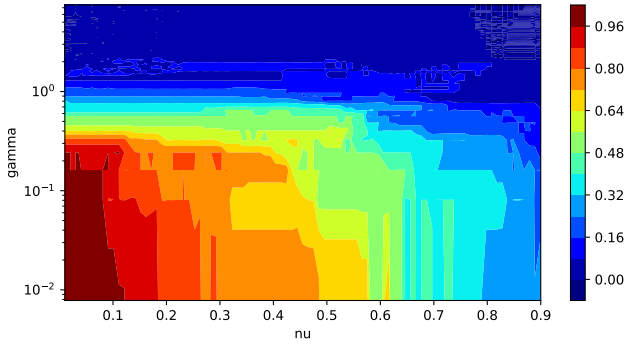

Figure 6.1: Contour plot depicting hyperparameter optimisation results for different combinations of (ν, γ) in the main engine condition monitoring case study. The Matthews Correlation Coefficient (MCC) score achieved in K-folding is depicted by the colour of the contours.

at +25°C.

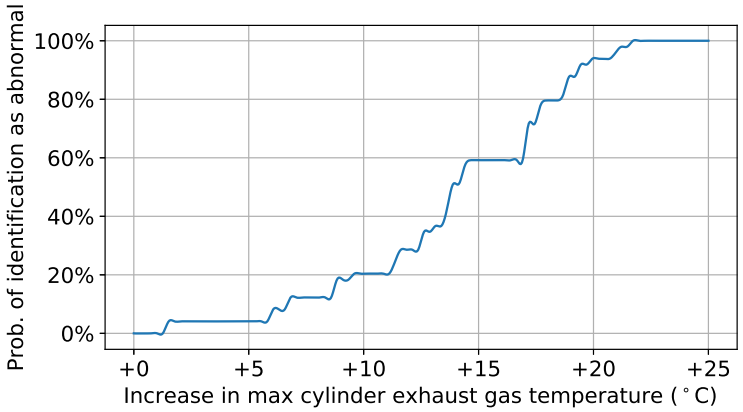

Figure 6.2: Max cylinder exhaust gas temperature sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for increasing max cylinder exhaust gas temperature measurements is displayed.

Accordingly, in Figure 6.4 model performance in an increase of standard deviation between exhaust gas temperature measurements of several cylinders is presented. Elevated exhaust gas temperature deviation is a known abnormality metric. Again, an increase of standard deviation of over +5°C returns a probability of abnormality of about 80%, with a probability of of 100% at +7°C.

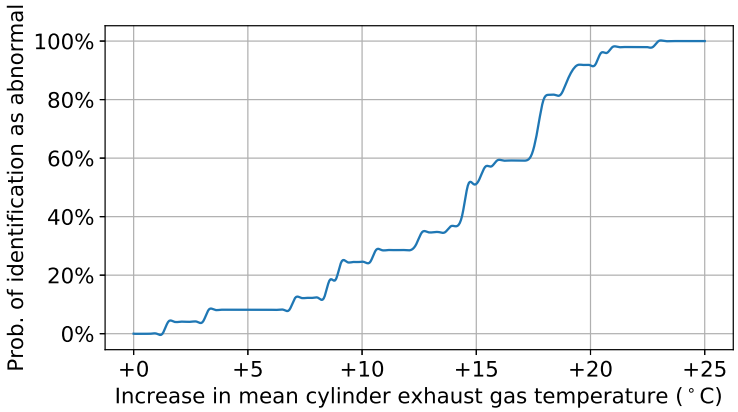

Figure 6.3: Mean cylinder exhaust gas temperature sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for increasing mean cylinder exhaust gas temperature measurements is displayed.

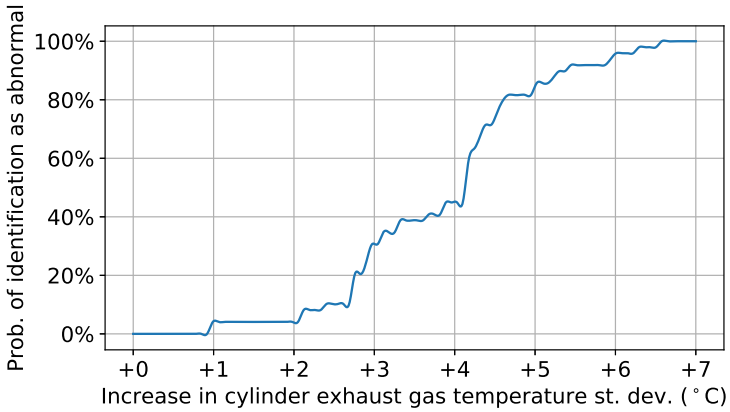

Figure 6.4: Cylinder exhaust gas temperature standard deviation sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for increasing cylinder exhaust gas temperature measurements standard deviation is displayed.

Figure 6.5 exhibits model performance while shaft RPM are dropping. While on its own, an RPM drop does not constitute a fault pattern but a mere operating profile shift, as all other measurements remain as in nominal condition at reduced RPM this scenario represents an abnormality. This can also be seen as a simulation where all other measurements are increased for constant RPM.

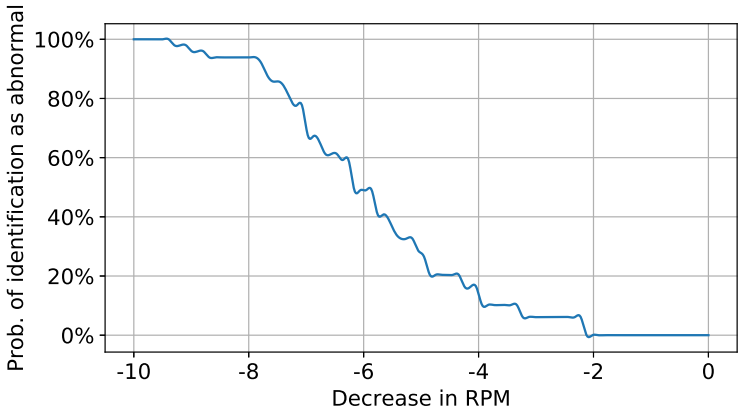

Figure 6.5: RPM sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for decreasing RPM measurements is displayed.

Figure 6.6 presents model performance as the differential between T/C inlet and outlet temperature decreases. A decrease in this differential corresponds to a reduction in T/C performance as less energy is extracted out of exhaust gas.

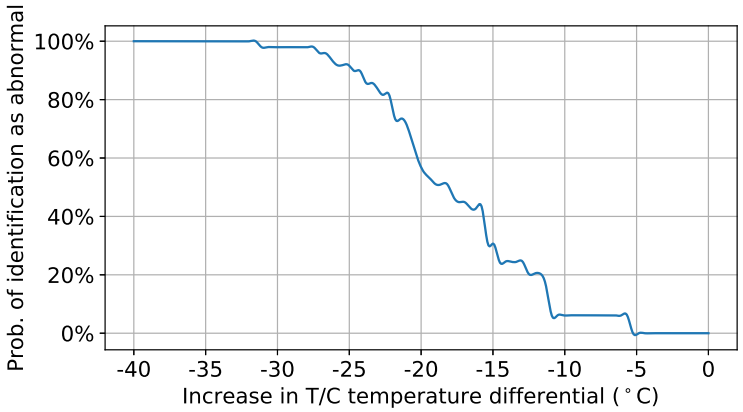

Figure 6.6: T/C inlet/outlet temperature differential sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for decreasing T/C inlet/outlet temperature differential measurements is displayed.

Figure 6.7 another interesting aspect of modelling at system level. Thrust bearing temperature measurement value is a variable that denotes abnormalities on its own, without requirement of modelling at system level. However, even in system-wide mod-

elling, deviating values are correctly detected, with an increasing probability of abnormality.

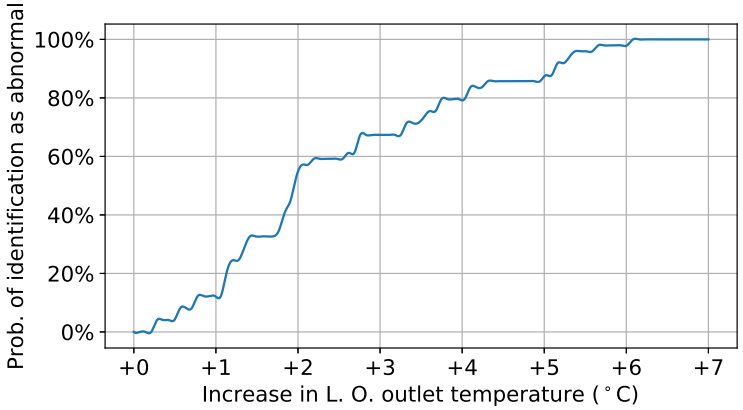

Figure 6.7: Thrust bearing temperature sensitivity analysis in the main engine dataset condition monitoring case study. Probability of the measurements being identified as abnormal for decreasing thrust bearing temperature measurements is displayed.

6.1.2 Auxiliary engine dataset AE1

Similarly to the above case studies, the discarding of 88 NaN values and outliers yielded 284 data points. Following the data splitting approach applied in the case study of Section 5.1.2, 199 observations are used for model training, 28 for validation and 57 for testing. Accordingly, the descriptive statistics of the training dataset of this case study is shown in Table 6.2.

Table 6.2: Descriptive statistics of the training part of the dataset used in the auxiliary engine AE2 dataset condition monitoring case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Power (kW)	L.O. inlet temp. (°C)	L.O. inlet press. (bar)	Cooling Fresh Water (CFW) inlet temp (°C)	EG temp. max (°C)	EG temp. min (°C)	AC outlet temp (°C)	AC inlet temp (°C)
count	199	199	199	199	199	199	199	199
mean	279.05	660.89	63.85	4.43	73.55	321.20	142.47	43.87
std	6.08	41.25	1.24	0.13	2.47	8.77	12.05	2.51
min	260.00	500.00	61.00	4.20	67.00	300.00	124.00	41.00
25%	280.00	650.00	63.00	4.30	72.50	320.00	134.00	42.00
50%	280.00	650.00	64.00	4.40	75.00	320.00	138.00	43.00
75%	280.00	700.00	65.00	4.50	75.00	330.00	158.00	46.00
max	300.00	750.00	66.00	4.70	77.00	330.00	164.00	49.00

In Figure 6.8, the results of of grid search employed to determine the optimal values of hyperparameters ν and γ in the case of Diesel Generator is shown. As in the first case

study, lighter colours correspond to better performance and conversely. Values towards the manually-imposed lower bounds of both hyperparameters seem to return better results, with optimal performance being obtained for $\nu = 0.250$ and $\gamma = 0.069$.

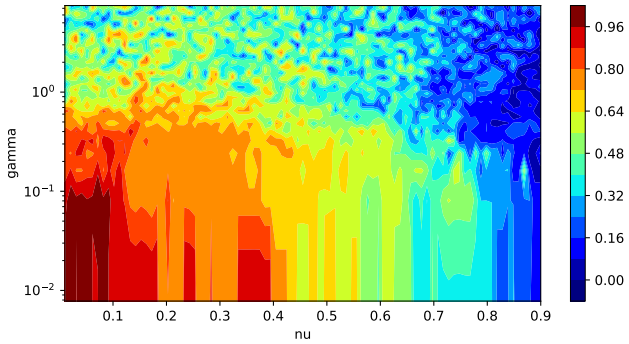

Figure 6.8: Contour plot depicting hyperparameter optimisation results for different combinations of (ν, γ) in the auxiliary engine AE2 dataset condition monitoring case study. The MCC score achieved in K-folding is depicted by the colour of the contours.

In Figure 6.9 model performance in increasing maximum exhaust gas temperature is presented. An increase of over +15°C returns a probability of abnormality of about 60%, with a probability of 100% at +30°C.

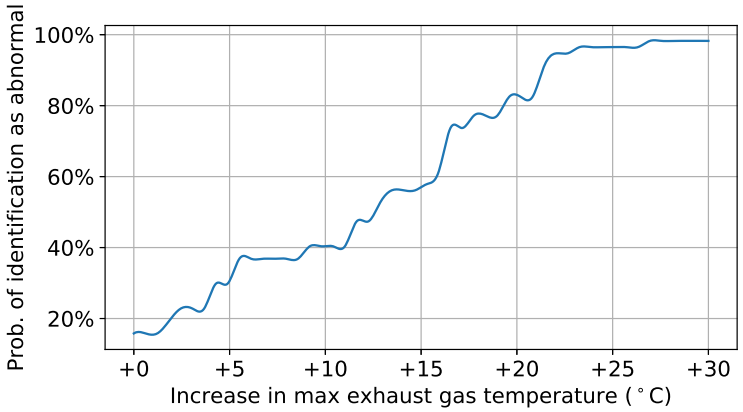

Figure 6.9: Max exhaust gas temperature sensitivity analysis in the auxiliary engine AE2 dataset condition monitoring case study. Probability of the measurements being identified as abnormal for increasing max exhaust gas temperature measurements is displayed.

In Figure 6.10 model performance at detecting anomalies when increasing L.O. outlet

temperature is presented. An increase of over +2°C returns a probability of abnormality of about 50%, with a probability of 100% at +5°C.

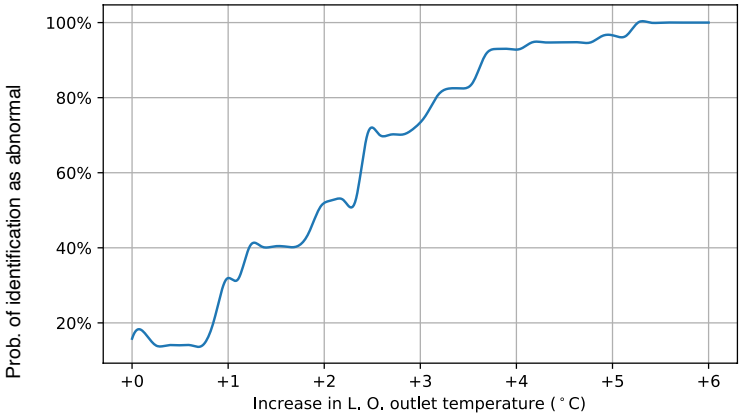

Figure 6.10: Thrust bearing LO temperature sensitivity analysis in the auxiliary engine AE2 dataset condition monitoring case study. Probability of the measurements being identified as abnormal for increasing thrust bearing LO temperature measurements is displayed.

6.1.3 Auxiliary engine dataset AE2

This section presents and discusses the results obtained through the pre-processing, model training, and relevant case studies introduced in Section 5.1.3.

As the auxiliary engine examined in this case study is not always operational, 291 points with NaN values are excluded from the analysis carried out, following the pre-processing methodology, as described in Section 4.3.2. Following that, 804 points remained available. Next, the outlier rejection step of this pre-processing methodology is applied, removing an additional 35 data points.

In order to ensure results fairness and avoid the introduction of unnecessary bias in the case study, the remaining dataset is split into training and testing before the training part is further examined to make certain that the testing part remains unseen. In this case, a $70\% - 30\%$ dataset split is selected for training and, respectively, testing. This is chosen arbitrarily, following data science standard practices that apply to relatively small datasets. In larger datasets, the amount of data points retained purely for testing purposes is reduced.

The descriptive statistics of the training dataset is shown in Table 6.3, where the mean, minimum and maximum values of each feature are presented, along with the standard deviation and the values at different (25, 50, and 75%) quantiles.

Table 6.3: Descriptive statistics of the training part of the dataset used in the auxiliary engine AE1 dataset condition monitoring case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Power (kW)	L.O. inlet temp. (°C)	L.O. inlet press. (bar)	CFW inlet temp (°C)	EG temp. max (°C)	EG temp. min (°C)
count	538	538	538	538	538	538
mean	652.23	64.05	4.50	72.34	305.45	285.69
std	51.94	1.48	0.14	2.92	13.20	12.82
min	500.00	60.00	4.10	64.00	270.00	260.00
25%	600.00	63.00	4.40	69.00	300.00	280.00
50%	650.00	64.00	4.50	73.00	300.00	280.00
75%	700.00	65.00	4.60	75.00	320.00	300.00
max	800.00	68.00	4.90	77.00	330.00	310.00

In Figure 6.11, the results of the random search employed to determine the optimal values of hyperparameters ν and γ in the model trained on the Diesel Generator set (D/Gen) dataset is presented, with a heatmap denoting the MCC values obtained in the validation phase and dots depicting the hyperparameter combinations that are tested. Darker colours correspond to better MCC performance on the validation dataset; and conversely. Lower values of ν, especially in the $(0, 0.2)$ range and γ values in the $(10^{-1}, 10^{-4})$ range yielded higher MCC values on validation. Higher MCC values indicate better model performance at accurately classifying data points as normal or abnormal. In order to obtain this heatmap, the One Class Classification (OCC) Support Vector Machine (SVM) model is trained for 3000 randomly selected (ν, γ) tuples from the relevant range discussed in Section 5.1.3. Through that, it is found that the optimal (ν, γ) tuple is (0.0049, 0.0016) for this case study.

In Figure 6.12, model performance in increasing CFW inlet temperature is presented. Normal behaviour (approximately first 230 points) of the CFW inlet temperature are associated within the $68 - 76\,°C$ range. Concurrently, the Exponential Weighted Moving Average (EWMA) of the model output is steadily positive, therefore correctly

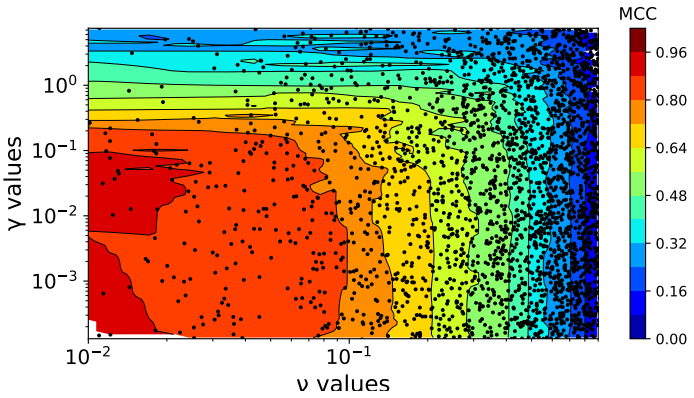

Figure 6.11: Contour plot depicting hyperparameter optimisation results for different combinations of (ν, γ) in the auxiliary engine AE1 dataset condition monitoring case study. Tested combinations are shown in dots and the MCC score achieved in K-folding is depicted by the colour of the contours.

identifying these points as normal.

As the CFW inlet temperature starts to increase, the distance of the relevant points from the hyperplane defining "normality" starts decreasing, eventually turning negative (i.e. points being classified as abnormal) slightly before the temperature crosses the alarm threshold. By the time the CFW inlet temperature reached the Original Equipment Manufacturer (OEM) shutdown threshold, the distance from hyperplane has taken a large negative value, with those points being labelled as abnormal with a high confidence. Based on the above results, it is inferred that the OCC SVM model implemented can accurately identify deviating patterns in a time-series dataset, and reflect that in its normality prediction.

In Figure 6.13, the same process is repeated for the L.O. inlet pressure measurement. Analysing the testing dataset, normal measurement range is within the $4.2 - 4.7$ bar range, with the OEM alarm threshold at 4.0 bar and the shutdown threshold at 3.5 bar.

As the L.O. pressure starts dropping, it is observed that the distance of the relevant points from the hyperplane decreases, and eventually becomes negative slightly before the pressure drops beyond the alarm threshold. This manifests the model's ability to

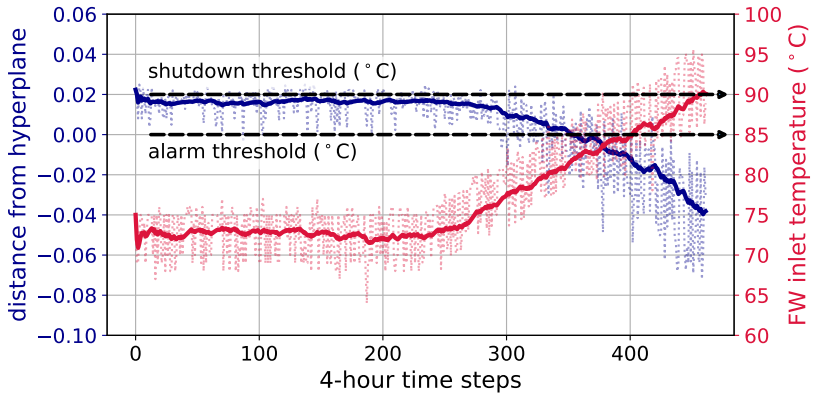

Figure 6.12: CFW inlet temperature sensitivity analysis in the auxiliary engine AE1 dataset condition monitoring case study. Actual distance from hyperplane (dark) and temperature measurements (light) are shown in thin, dotted lines and the EWMA-filtered values in a thick, solid line. Positive distance from hyperplane signifies points identified as "normal"; and conversely. A larger distance conveys a higher prediction confidence.

identify incipient deviations from the "normal" pattern that is learnt during training. Similarly to the preceding CFW inlet temperature case study, as the pressure reaches the OEM shutdown threshold, the corresponding points are assigned a large negative distance value.

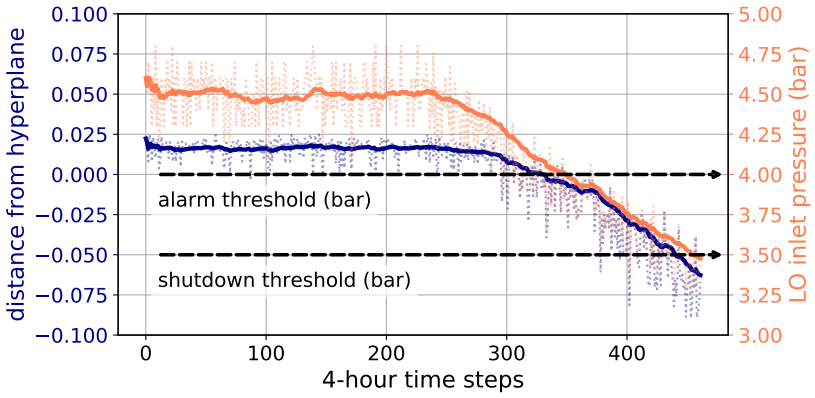

Figure 6.13: L.O. inlet pressure sensitivity analysis in the auxiliary engine AE1 dataset condition monitoring case study. Actual distance from hyperplane (dark) and pressure measurements (light) are shown in thin, dotted lines and the EWMA-filtered values in thick, solid line. Positive distance from hyperplane signifies points identified as "normal"; and conversely. A larger distance conveys a higher prediction confidence.

Complimentary to the L.O. inlet pressure sensitivity analysis presented above, the

same methodology is repeated for the L.O. inlet temperature in Figure 6.14. For this measurement, the OEM only provides the shutdown threshold, at 75°C and normal measurement range at the $62 - 66$°C range.

In this case, it can be observed that as the L.O. inlet temperature increases towards the shutdown threshold, the distance from the separating hyperplane becomes increasingly negative. Following the results of the previous two case studies where it is observed that the alarm threshold is crossed at values where the hyperplane distance is approximately zero, a reversed process could be employed to identify an alarm threshold at approximately 70°C for L.O. inlet temperature measurements.

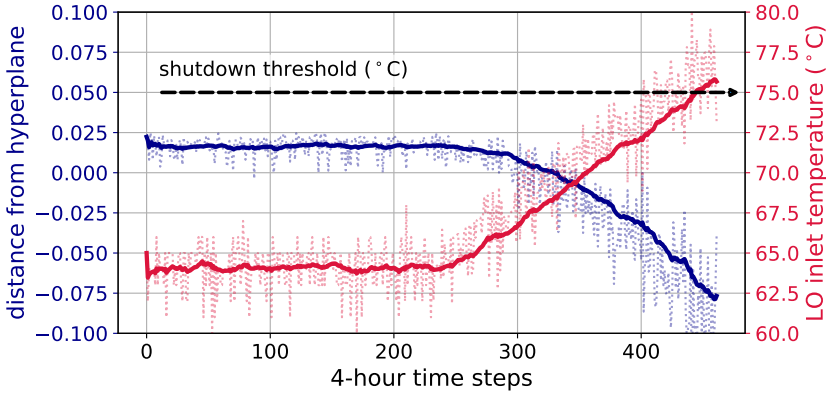

Figure 6.14: L.O. inlet temperature sensitivity analysis in the auxiliary engine AE1 dataset condition monitoring case study. Actual distance from hyperplane (dark) and temperature measurements (light) are shown in thin, dotted lines and the EWMA-filtered values in a thick, solid line. Positive distance from hyperplane signifies points identified as "normal"; and conversely. A larger distance conveys a higher prediction confidence.

Finally, in Figure 6.15, the sensitivity analysis for the maximum cylinder exhaust port gas temperature measurement is demonstrated. This measurement is utilised as the measurements for all cylinders are aggregated on board the vessel and only their max/min values are transported on shore. OEM provides an overall exhaust gas temperature alarm threshold at 480°C while measurement range observed in the testing dataset is at $300 - 320$°C.

In this case, it is observed that as the maximum exhaust gas temperature increases, the

distance from hyperplane decreases and becomes negative. However, at this case study the deviation that triggers the distance-from-hyperplane drop is detected at approximately two-thirds of the actual alarm threshold. This presents the main shortcoming of the proposed methodology, i.e. the fact that the normality predicted by the model is affected by the range of inputs at the training stage. This effect is mostly present in the case of the D/Gen exhaust gas temperature, as this measurement correlates majorly with engine operating conditions whereas all other measurements examined fluctuate in a OEM preset range through the operation of coolers and pumps. If only a subset of normal operating conditions are provided, the model will focus on that range, identifying anything beyond that as abnormal. However, this can be easily resolved by re-training the model when the range of normal operating conditions is modified.

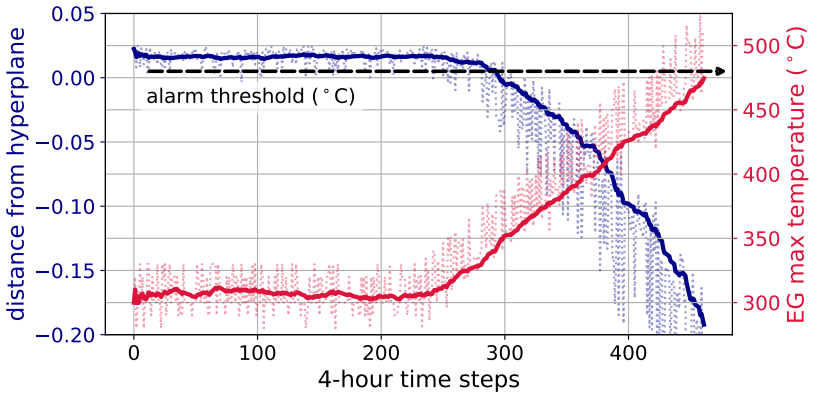

Figure 6.15: Maximum exhaust gas temperature sensitivity analysis in the auxiliary engine AE1 dataset condition monitoring case study. Actual distance from hyperplane (dark) and temperature measurements (light) are shown in thin, dotted lines and the EWMA-filtered values in a thick, solid line. Positive distance from hyperplane signifies points identified as "normal"; and conversely. A larger distance conveys a higher prediction confidence.

6.1.4 Key findings

This case study is based on the novel condition monitoring methodology described in Section 4.5, for the monitoring and detection of operating anomalies in ship machinery using measured process data. For this, an OCC SVM model is trained using a dataset corresponding to nominal behaviour of marine machinery under varying operating

conditions, with modelling selections being extensively discussed. The main findings of this application are summarised below:

The developed models are able to accurately discern between nominal and abnormal machinery condition, providing a suitable basis for an incipient-fault detection system. Furthermore, due to limited amount of assumptions, this methodology can be applied to a diverse set of machinery. This is corroborated by the fact that in the three applications presented above, different systems and sensor data are used depending on the inherent data availability, without affecting the accuracy of the models.

Contrary to other condition monitoring approaches, model training only requires reference (i.e. "nominal") data, majorly simplifying the process. Data requirements can be satisfied from noon-report data, significantly decreasing, or even avoiding, Capital Expenditure (CAPEX) costs traditionally attached to condition monitoring applications. Moreover, the knowledge and experience accumulated by operators can be leveraged to ensure the normality of the points fed to the model at training time. In this case, the term "normal" is situation- and operator-specific as it relates, among others, to different vessel baseline conditions and maintenance budgets.

Accordingly, comparing the proposed OCC ML methodology with conventional statistical techniques, this approach allows the prediction of the current condition whereas traditional statistial techniques would only have descriptive power, without the ability to provide insights. In the same vein, given that this approach is based on the derivation of a suitable hyperplane that encapsules normal data, adapting the model to account for changes in what is considered normal due to overall vessel deterioration can be resolved by retraining the model.

Besides, using the EWMA filter, the underlying methodology becomes dynamic as previous data points are taken into consideration when predicting the normality of new data points. Through the performed sensitivity analysis, model alarm thresholds can be obtained. Comparing these values with the ones provided by OEMs, a significant degree of similarity is observed. Even in cases where only a limited amount of OEM thresholds

are available, relevant thresholds can be suggested by analysing model output.

6.2 FOC modelling comparison

This section presents and discusses the results obtained from the case study originally described in Section 5.2. The aim of this case study is the identify optimal FOC modelling approaches when different data sources exist, namely noon reports and Automated Data Logging & Monitoring (ADLM) systems.

The same amount of training points are selected from both datasets, in order to keep them similar, thus avoiding a biased comparison. Aiming for an approximately 80-20% split in training and testing data, for both cases 603 training points are retained, with the remaining 20% being used for testing. Following their pre-processing, the two datasets are presented below, in Sections 6.2.1 and 6.2.2.

6.2.1 Noon-report (V1) dataset description

In regards to the V1 dataset, 834 data points were originally made available from a 1100 TEU reefer vessel, corresponding to approximately 2.5 years of noon-report data. Following pre-processing, 745 points were contained in the dataset.

Histograms of the measured parameters used for model training are shown in Figure 6.16. Histograms constitute a discretised representation of the distribution of variables using bars of different sizes. The width of each bar describes the range of each bin. Accordingly, the horizontal axis of each plot denotes the number of points corresponding to each histogram bin. For example, in the case of vessel speed, we observe a negatively skewed distribution where most observations lie close to 15 knots with very few observations having a value below 10 knots.

Furthermore, the descriptive statistics of this dataset are shown in Table 6.4. Considering the case of Main Engine (M/E) Speed as an example, the 745 measurements contained in the dataset have a mean value of 110.93 RPM with a standard deviation of 11.71 RPM. Moreover, their minimum observed value is at 47.40 RPM whilst their

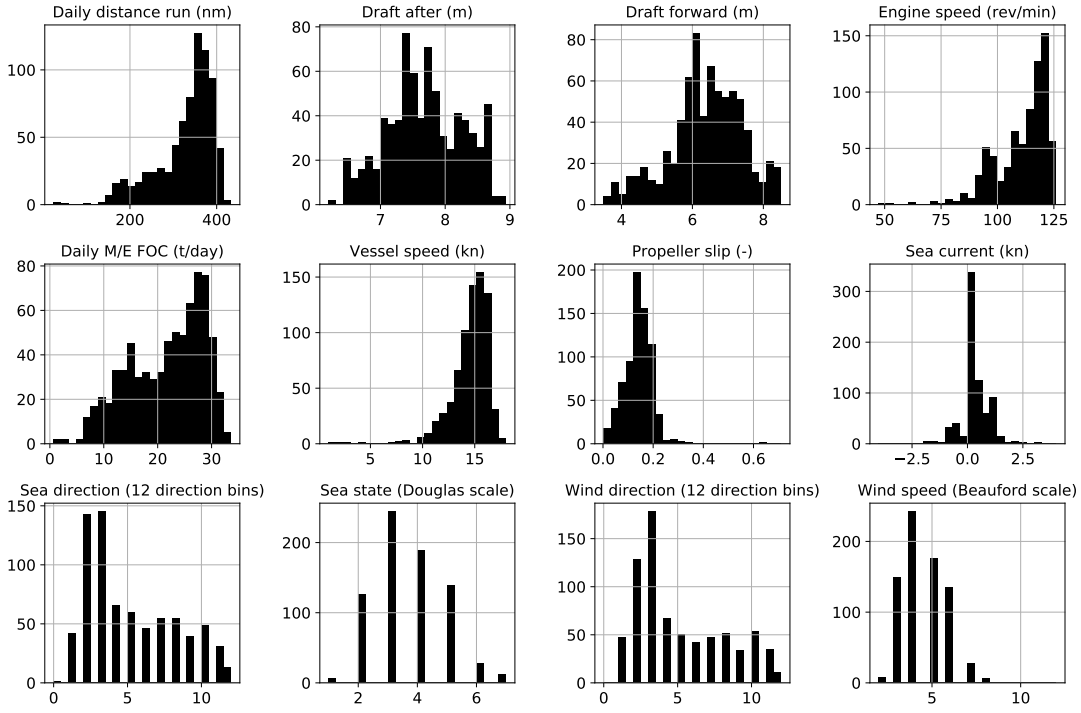

Figure 6.16: Histogram plots of the training part of the dataset used in the noon-report (V1) dataset FOC modelling comparison case study. The horizontal axis of each plot denotes the number of points corresponding to each histogram bin.

maximum observed value is at 126.00 RPM. Furthermore, 25% of the speed observations have a value below 104.70 RPM, and 75% of the observations have a value below 120.00 RPM. The median value of the dataset, i.e. the value corresponding to the fiftieth percentile, is at 115.10 RPM.

Additionally, as a brief investigation of the available training dataset, a correlation matrix is obtained, focusing on the daily FOC correlation to other measured parameters, shown in Table 6.5. Correlation matrices express the relationship between available quantities. As in this case, FOC is the independent variable, the relationship between all other quantities and FOC is examined. It is important to note that a correlation matrix, and correlation in general, only expresses the amount of linear relationship between two variables; any non-linear connection will not be captured by this. In this case, engine speed and vessel speed exhibit the highest correlation with vessel FOC. Ambient weather conditions exhibit a limited correlation with vessel FOC. Finally, ves-

Table 6.4: Descriptive statistics of the training part of the dataset used in the noon-report (V1) dataset FOC modelling comparison case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Distance run (nm)	Draft aft (m)	Draft forward (m)	M/E Speed (RPM)	M/E FOC (tn/day)	Vessel Speed (kn)	Propeller Slip (-)	Sea Current (kn)	Sea Direction (0-12)	Sea Force (0-12)	Wind Direction (0-12)	Beaufort wind force (0-12)
count	745	745	745	745	745	745	745	745	745	745	745	745
mean	327.32	7.67	6.38	110.93	21.68	14.56	0.14	0.26	5.06	3.62	4.98	4.53
σ	66.11	0.58	1.03	11.71	6.92	1.82	0.06	0.68	3.07	1.19	3.09	1.21
min	24.00	6.20	3.50	47.40	0.70	1.00	0.00	-4.00	0.00	1.00	0.00	2.00
25%	301.00	7.30	5.85	104.70	15.80	13.87	0.11	0.00	3.00	3.00	3.00	4.00
50%	349.00	7.65	6.48	115.10	23.10	14.88	0.14	0.00	4.00	3.00	4.00	4.00
75%	374.60	8.17	7.10	120.00	27.50	15.75	0.17	0.60	8.00	4.00	7.00	5.00
max	433.00	8.94	8.50	126.00	33.60	18.00	0.71	4.00	12.00	7.00	12.00	12.00

sel drafts do not seem to exhibit a perceivable correlation with vessel FOC. This is due to the fact that this vessel has relatively limited drafts range due to the inherent voyage characteristics of containerships. This claim is corroborated by the draft histograms depicted in Figure 6.16.

Table 6.5: Correlation of FOC to other measured attributes in the training part of the dataset used in the noon-report (V1) dataset FOC modelling comparison case study. Positive values denote a positive linear correlation; negative values denote a negative linear correlation; and very small values denote lack of linear correlation.

Attribute	**Correlation coefficient (-)**
M/E speed (RPM)	0.84
Speed (kn)	0.59
Sea state	0.32
Wind speed (m/s)	0.31
Propeller slip	0.14
Draft aft (m)	0.09
Draft forward (m)	-0.03
Sea current (kn)	-0.08
Wind direction	-0.19
Sea direction	-0.19

6.2.2 ADLM (V2) dataset description

In this second data set, 922 hourly-collected data points acquired from a bulk carrier of approximately 200,000 DWT equipped with an ADLM system are analysed. 768 data points remained in the dataset following the application of the pre-processing methodology. In this case, while ADLM systems usually provide a wider range of measured parameters, the same parameters as in the case of the noon-report data are considered. This decision is made so that the study focuses on the quality benefits of the

data provided by the different data acquisition strategies. The descriptive statistics of this dataset are presented in Table 6.6. Following the paradigm set in the description of dataset V1, the descriptive statistics of the M/E speed will be elaborated as an example. In this dataset, the mean engine speed value is 62.76 RPM with a standard deviation of 6.70 RPM. The minimum value observed is 12.90 RPM whilst the maximum observed value is at 68.58 RPM. 25% of the speed observations have a value below 58.73 RPM, and 75% of the observations have a value below 67.57 RPM. The median value of the dataset is at 66.99 RPM.

However, given that in this case, data are provided at an hourly sampling rate, in order to retain the same amount of training and testing points so that an unbiased comparison of the two data acquisition strategies is made, a three-month time window is selected. Filtering conditions are similar to the ones presented above, without the need to filter for daily steam hours. Following this filtering process, 603 points are retained and used for training.

Table 6.6: Descriptive statistics of the training part of the dataset used in the ADLM (V2) dataset FOC modelling comparison case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Distance run (nm)	Draft aft (m)	Draft forward (m)	M/E Speed (RPM)	M/E FOC (tn/day)	Vessel Speed (kn)	Propeller Slip (-)	Sea Current (kn)	Sea Direction (degrees)	Sea Force (m)	Wind Direction (degrees)	Wind Speed (m/s)
count	768	768	768	768	768	768	768	768	768	768	768	768
mean	283.23	14.24	13.67	62.76	38.22	11.80	0.09	-0.46	189.92	0.55	136.38	20.94
σ	43.20	4.46	5.10	6.70	7.86	1.80	0.08	0.77	123.42	0.65	135.03	9.39
min	26.75	7.22	6.73	12.90	5.02	1.11	0.00	-4.85	0.05	0.00	2.24	0.41
25%	268.80	9.00	7.60	58.73	30.74	11.20	0.06	-0.92	64.91	0.00	20.34	13.64
50%	293.64	17.40	17.45	66.99	43.02	12.23	0.08	-0.44	171.68	0.30	49.99	20.76
75%	308.34	18.25	18.20	67.57	44.58	12.85	0.10	0.02	311.73	1.02	298.83	27.58
max	386.05	18.30	18.30	68.58	54.17	16.09	0.70	2.47	359.82	3.05	350.07	52.53

A correlation matrix is also derived for this case, providing insight on how the daily FOC correlates with the other measured parameters, shown in Table 6.7. Evaluating against Table 6.5, it is inferred that in both cases the M/E speed (and, therefore, also vessel speed) is identified as significantly correlated with the FOC of the vessel. However, whilst in this case, both draft forward and aft have a high increased correlation with FOC, which is not the case for dataset V1. This is mostly due to the fact that, as can be seen in Figure 6.17, vessel V2 essentially only takes two draft values: one for ballast condition and one for laden conditions, whereas vessel V1 operates in a wide range of

draft values. This is equally reflected in the FOC and engine speed histograms where, in both cases, a bimodal distribution is observed. This can be explained by the fact that vessel V1 is a reefer vessel, going from port to port and loading/unloading containers; rarely being neither at full load nor at ballast condition. At the same time, vessel V2 is a large bulk carrier, always leaving at full load from the departure port, heading to to destination to unload all cargo, and then moving at ballast condition from that port to another, where she will load cargo again; essentially alternating between these two load conditions.

Table 6.7: Correlation of FOC to other measured attributes in the training part of the dataset used in the ADLM (V2) dataset FOC modelling comparison case study. Positive values denote a positive linear correlation; negative values denote a negative linear correlation; and very small values denote lack of linear correlation.

Attribute	**Correlation coefficient (-)**
M/E speed (RPM)	0.89
Draft aft (m)	0.84
Draft forward (m)	0.84
Speed (kn)	0.57
Sea direction	-0.05
Propeller Slip	0.00
Wind direction	-0.05
Sea state	-0.05
Sea current (kn)	-0.06
Wind speed (m/s)	-0.16

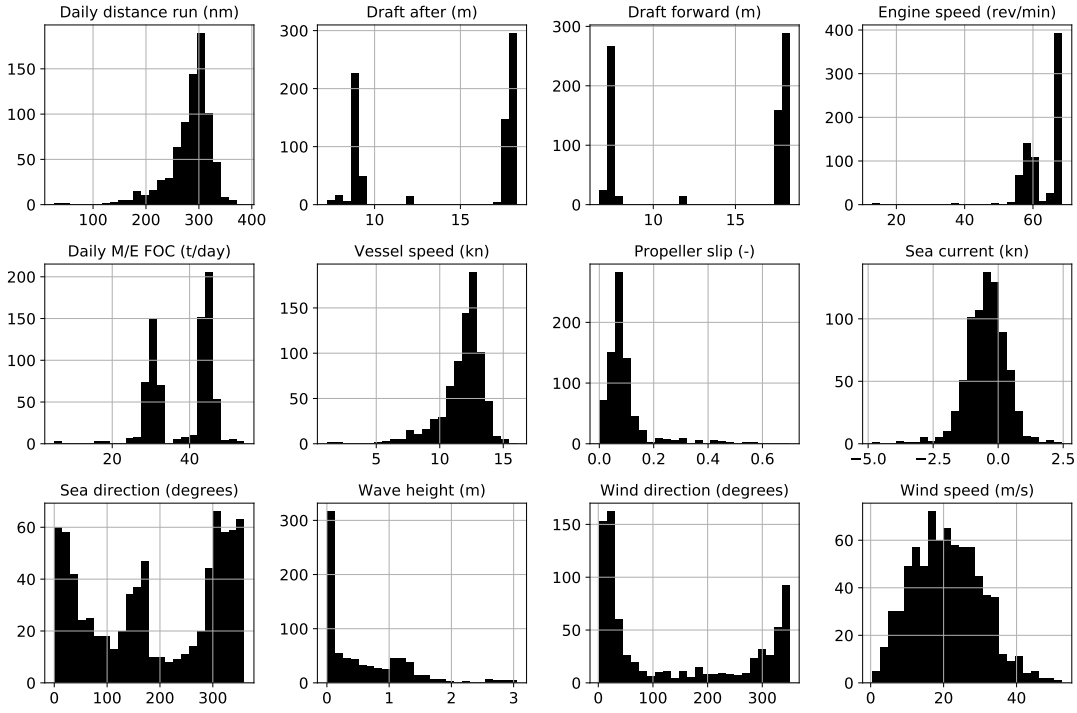

Figure 6.17: Histogram plots of the training part of the dataset used in the ADLM (V2) dataset FOC modelling comparison case study. The horizontal axis of each plot denotes the number of points corresponding to each histogram bin.

6.2.3 Models training

The times that each model took to train with and without hyperparameter optimisation are presented in Table 6.8. Training is repeated 10 times for each dataset in order to get consistent results, which are then averaged for the two datasets in order to get one common, consistent result. Both datasets exhibited similar training times, which is expected given that both contain the same number of training points. In all cases where no hyperparameter tuning was applied, a training time of less than 1 minute was observed. Conversely, when hyperparameter tuning was applied, tree-based methods and ANNs exhibited increased training times. The training time of SVRs and LR-based methods was not significantly increased due to hyperparameter optimisation.

Table 6.8: Model training time with and without hyperparameter optimisation for the FOC modelling comparison case study. Training is repeated 10 times for each dataset in order to get consistent results, which are then averaged for the two datasets in order to get one common, consistent result.

Model	**Training time (min.)**	**Hyperparameters optimisation?**
Linear Regression (LR)	< 1	✗
Least Absolute Shrinkage and Selection Operator (LASSO)	< 1	✗
	1	✓
Ridge Regression (RR)	< 1	✗
	2	✓
Elastic net	< 1	✗
	1	✓
Decision Tree Regressor (DTR)	< 1	✗
	8	✓
Random Forest Regressor (RFR)	< 1	✗
	34	✓
K-Nearest Neighbours (KNN)	< 1	✗
	2	✓
Support Vector Reggressor (SVR)	< 1	✗
	2	✓
Extra Trees Regressor (ETR)	< 1	✗
	42	✓
Artificial Neural Network (ANN)	< 1	✗
	38	✓

6.2.4 Noon-report (V1) dataset results

An overview of the obtained results is presented in the box plots of Figure 6.18. The line inside each box corresponds to the median (second quartile) score of this model in k-folding, the top and bottom of the box respectively correspond to the first and third quartiles. The whiskers represent the lowest point of data within 1.5 Interquartile Range (IQR) of the lowest quartile and the highest point of data within 1.5 IQR of the upper quartile. Accordingly, the mean of the dataset is noted by a triangle. Data points beyond the whisker range as shown individually in the form of hollow circles.

This figure shows the extent of the effectiveness of the default hyperparameters included in Scikit-learn, as in many cases only a minuscule gain in R^2 is obtained after the hyperparameter optimisation loop. Additionally, most modelling attempts delivered overall good results, with a mean/median R^2 of over 85% in most cases. In this case, RFRs yielded the best results, closely followed by ANNs of 1- and 2-layers and the SVR

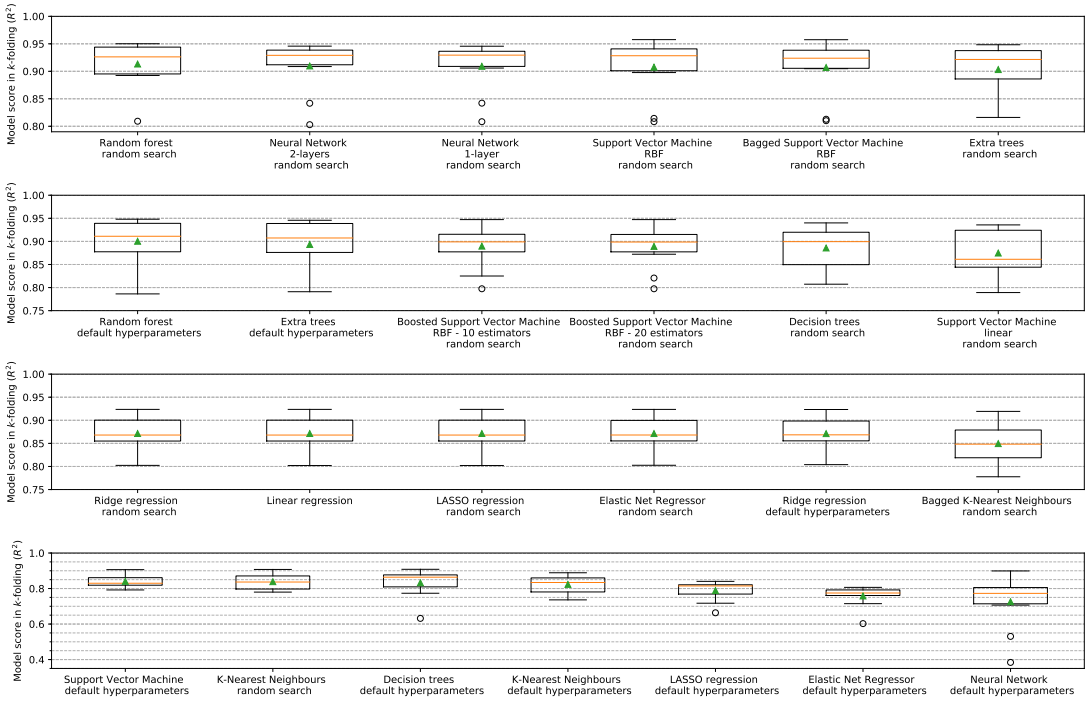

Figure 6.18: Box plot of the R^2 obtained from different models and hyperparameters in K-folding in the noon-report (V1) dataset FOC modelling comparison case study. The line inside each box corresponds to the median (second quartile) score of this model in k-folding, the top and bottom of the box respectively correspond to the first and third quartiles. The whiskers represent the lowest point of data within 1.5 IQR of the lowest quartile and the highest point of data within 1.5 IQR of the upper quartile. Accordingly, the mean of the dataset is noted by a triangle. Data points beyond the whisker range as shown individually in the form of hollow circles.

with the Radial Basis Function (RBF) kernel. Therefore, in this case, the evaluation of different models should be prioritised against a thorough hyperparameter optimisation. The only case where this is not true is in the case of regularised LR (i.e. LASSO, Ridge, and Elastic Net regressors), where the default regularisation term yielded suboptimal results. At the same time, due to the relatively small dataset size and lack of outliers, regularisation did not yield any improvement, with the unregularised LR providing some of the best results for this category of modelling techniques. Regarding ensemble techniques, in the case of SVRs, bagging provided better results than boosting, but a single regressor still provided a better mean R^2, albeit with a slightly increased variance.

Another way of evaluating results is through their achieved R^2 when the number of training points is altered. This is visualised in Figure 6.19, showing both training and cross-validation R^2 along with their respective 95% confidence intervals for the top-six models derived.

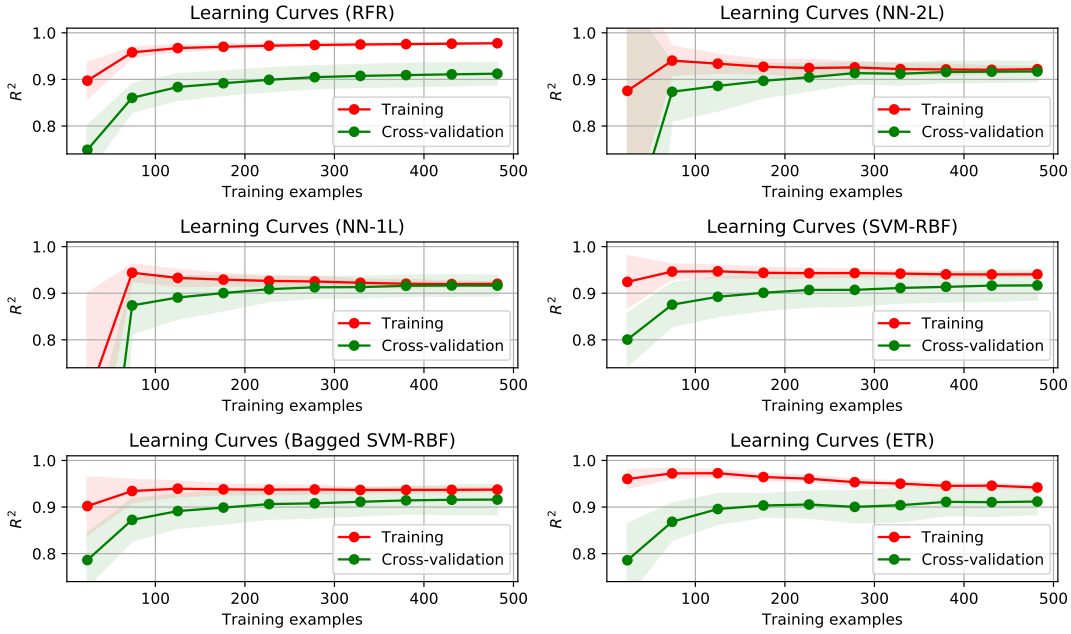

Figure 6.19: Training curves for the models that achieved best cross-validation R^2 in the noon-report (V1) dataset FOC modelling comparison case study. The lightly tinted areas denote the 95% confidence interval. In cases where two tinted areas overlap, a darker shade is displayed. Training curves display the obtained R^2 score in training and cross-validation for increasing number of training points.

95% confidence intervals define the area that the true R^2 value will lie with a probability of 95%. As in training the same dataset is intrinsically used to obtain the model parameters and the train R^2 score, very small confidence intervals are obtained, i.e. more precise estimates. In the case of cross-validation, larger confidence intervals are expected as, in this case, the generalisation capabilities of the model are evaluated.

All six models achieve an R^2 of 85 to 90% when approximately 80 points are used, increasing to over 90% for more points. At the same time, the RFR and ETR models exhibit significantly reduced uncertainty compared to other models, especially when a small number of points is used. After approximately 300 points, all models seem to plat-

eau, without any tangible increase in model performance when additional model points are included. Nevertheless, the RFR model exhibited high variance that can be seen in the large gap that is present between the training and cross-validation curves. This signifies that while RFR presented the best results, there also exist a minor tendency to model the random noise in the data additionally to the actual features (overfitting).

Having identified that RFR overall exhibited the best R^2 in K-folding, the same parameters are now tested in the dataset held aside for validation. There, an $R^2 = 88.5\%$ is obtained, along with a mean error of 1.45 t/day and a median error of 1.0 t/day.

While normally this would be the only model that would be evaluated against the testing dataset, in the case of this investigation, the performance of all models is included in Table 6.9. This is due to the fact that the aim of this case study is to investigate the performance of different models whilst obtaining useful insights and not necessarily to derive a single model to model the FOC of this specific vessel.

Table 6.9: Testing scores of models considered in the noon-report (V1) dataset FOC modelling comparison case study. Best-performing models are presented in bold.

		Expl. variance (%)	MeanAE (tn/day)	MSLE ($\log(\text{tn/day})^2$)	MSE ($(\text{tn/day})^2$)	MedianAE (tn/day)	R^2 (%)
Linear Regression		87.25	1.674	0.050	6.108	1.217	86.79
LASSO	Default	74.94	2.742	0.075	12.544	2.463	72.86
	Hyperparameter optim.	87.25	1.674	0.050	6.108	1.217	86.79
Ridge	Default	87.22	1.677	0.050	6.125	1.208	86.75
	Hyperparameter optim.	87.25	1.674	0.050	6.110	1.216	86.78
Elastic Net	Default	71.80	2.888	0.080	13.815	2.676	70.11
	Hyperparameter optim.	87.24	1.676	0.050	6.115	1.218	86.77
Decision trees	Default	78.35	2.099	0.056	10.174	1.400	77.99
	Hyperparameter optim.	84.44	1.836	0.052	7.557	1.271	83.65
Random Forests	Default	87.85	1.506	0.049	5.753	0.995	87.55
	Hyperparameter optim.	88.75	1.454	0.047	5.297	1.009	88.54
KNN	Default	81.42	2.227	0.066	9.834	1.590	78.73
	Hyperparameter optim.	77.60	2.419	0.073	12.272	1.862	73.45
SVM	Default	88.08	1.521	0.047	5.576	1.025	87.94
	Hyperparameter optim. (RBF)	**91.52**	**1.226**	**0.042**	**3.950**	**0.817**	**91.46**
	Hyperparameter optim. (linear)	88.44	1.566	0.049	5.506	1.066	88.09
Extra trees	Default	89.65	1.405	0.045	4.944	0.840	89.31
	Hyperparameter optim.	88.89	1.434	0.047	5.192	1.011	88.77
Boosting	SVR ×10	90.77	1.315	0.043	4.311	0.911	90.67
	SVR ×20	90.94	1.299	0.042	4.242	0.910	90.82
Bagging	KNN	81.62	2.198	0.065	9.996	1.560	78.37
	SVR	91.02	1.292	0.043	4.218	0.886	90.87
ANN	Default	85.02	1.892	0.054	7.068	1.411	84.71
	Hyperparameter optim. (1-layer)	89.52	1.414	0.043	4.869	0.945	89.47
	Hyperparameter optim. (2-layer)	88.99	1.432	0.044	5.121	0.926	88.92

Through Table 6.9, it is observed that in the testing dataset ANNs performed slightly better than RFRs, obtaining an R^2 increased by approximately 0.75%. Furthermore,

the RBF-based SVR obtained an even increased R^2 at approximately 91.50%. This discrepancy can be justified by the increased variance that this model exhibited at the validation stage.

6.2.5 ADLM (V2) dataset results

Following the model training, the derived R^2 coefficient of each model is visualised in Figure 6.20. From the results presented, it can be deduced that the best performing model is the ETR, with hyperparameter optimisation through random search, achieving an average coefficient of determination (R^2) of 97.7% and a median of over 98%. Random forests also yielded a comparable R^2, followed by 1-, and 2-layer Neural Networks.

The learning curves of this dataset are presented in Figure 6.21. Similarly to the results of Section 6.2.4, in training very small confidence intervals are obtained, denoting more precise estimates. In the case of cross-validation, larger confidence intervals are expected as, in this case, the generalisation capabilities of the model are evaluated. Even in the second case, high R^2 with relatively small confidence intervals are observed, especially in the cases where more than 300 examples are considered for model training.

ETR, RFR, and 1-layer ANN models performed similarly, with an R^2 of approximately 90% when 100 training points are used. As the number of points increases, the cross-validation R^2 is asymptotically approaching 100%. Nevertheless, in the case of the 2-layer ANN a high confidence interval is obtained, coupled with a lacking performance when a small amount of points is present. At the same time, when a larger number of points is used for training, a promising slope is present in the cross-validation R^2 curve, signifying that the R^2 coefficient may be increased for cases where an even larger number of training points is used.

Following the selection of ETR as optimal model due to providing the highest mean R^2 value, this model is evaluated on the testing dataset to ensure its generalisation

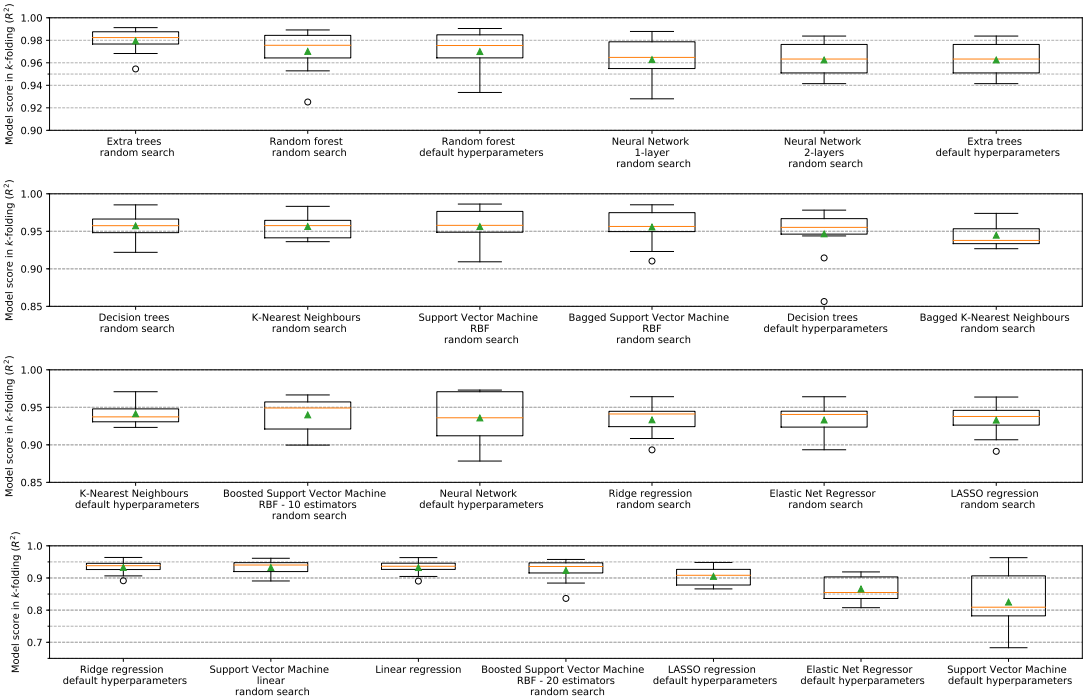

Figure 6.20: Box plot of the R^2 obtained from different models and hyperparameters in K-folding in the ADLM (V2) dataset FOC modelling comparison case study. The line inside each box corresponds to the median (second quartile) score of this model in k-folding, the top and bottom of the box respectively correspond to the first and third quartiles. The whiskers represent the lowest point of data within 1.5 IQR of the lowest quartile and the highest point of data within 1.5 IQR of the upper quartile. Accordingly, the mean of the dataset is noted by a triangle. Data points beyond the whisker range as shown individually in the form of hollow circles.

capabilities in previously-unseen input. In this case, ETR obtained an R^2 of 97.3%, with a mean error of 0.5 t/day and a median error of 0.2 t/day.

As discussed in the previous subsection, the performance of all models is included in Table 6.10. Here, is inferred that ETR yielded the best results across all metrics. At the same time, RFR models also yielded comparable performance. However, the employed ANN models that performed exceptionally well at validation, did not perform equally well at the testing stage yielding a mean average error of over 0.9 t/day.

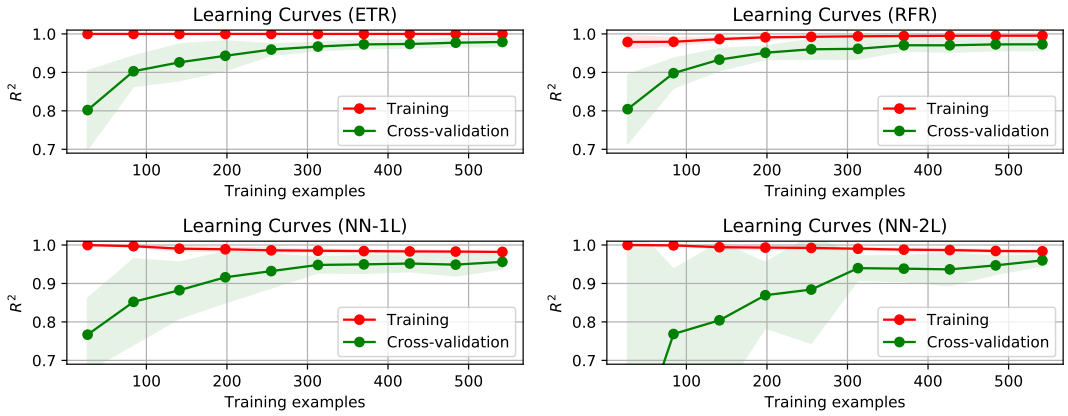

Figure 6.21: Training curves for the models that achieved the best R^2 at training for dataset V2. The lightly tinted areas denote the 95% confidence interval. In cases where two tinted areas overlap, a darker shade is displayed. Training curves display the obtained R^2 score in training and cross-validation for increasing number of training points.

Table 6.10: Testing scores of models considered in the ADLM (V2) dataset FOC modelling comparison case study. Best-performing models are presented in bold.

		Expl. variance (%)	MeanAE (tn/day)	MSLE ($\log(\text{tn/day})^2$)	MSE ($(\text{tn/day})^2$)	MedianAE (tn/day)	R^2 (%)
Linear Regression		94.49	1.409	0.006	3.670	1.106	94.48
LASSO	Default	93.77	1.331	0.005	4.151	0.715	93.76
	Hyperparameter optim.	94.51	1.410	0.006	3.658	1.134	94.50
Ridge	Default	94.48	1.413	0.006	3.671	1.144	94.48
	Hyperparameter optim.	94.64	1.404	0.005	3.570	1.203	94.63
Elastic Net	Default	89.95	1.732	0.014	6.696	1.091	89.93
	Hyperparameter optim.	94.69	1.398	0.005	3.532	1.201	94.69
Decision trees	Default	95.13	0.679	0.003	3.255	0.269	95.11
	Hyperparameter optim.	94.64	0.844	0.005	3.569	0.318	94.63
Random Forests	Default	96.27	0.570	0.003	2.487	0.192	96.26
	Hyperparameter optim.	96.38	0.564	0.003	2.405	0.234	96.38
KNN	Default	95.83	0.943	0.004	2.802	0.484	95.79
	Hyperparameter optim.	95.90	0.675	0.002	2.739	0.242	95.88
SVM	Default	73.11	1.843	0.039	18.161	0.809	72.69
	Hyperparameter optim. (RBF)	95.98	0.895	0.003	2.713	0.465	95.92
	Hyperparameter optim. (linear)	94.49	1.445	0.005	3.672	1.325	94.48
Extra trees	Default	96.22	0.756	0.008	2.522	0.311	96.21
	Hyperparameter optim.	**97.31**	**0.534**	**0.002**	**1.804**	**0.178**	**97.29**
AdaBoost	SVR ×10	95.89	1.142	0.004	2.737	0.740	95.89
	SVR ×20	95.23	1.240	0.004	3.171	0.911	95.23
Bagging	KNN	95.26	0.966	0.006	3.210	0.454	95.17
	SVR	95.94	0.925	0.003	2.751	0.467	95.86
ANN	Default	96.96	0.903	0.007	2.035	0.501	96.94
	Hyperparameter optim. (1-layer)	90.87	1.314	0.016	6.081	0.699	90.86
	Hyperparameter optim. (2-layer)	95.58	0.939	0.006	2.984	0.468	95.51

6.2.6 Key findings

This case study presents the application of the novel FOC modelling comparison of Section 4.6. This methodology aims to compare modelling approaches and data sources to derive an optimal approach for the prediction of M/E FOC of sailing vessels.

Based on the results of this case study, the derived models can accurately predict the FOC of vessels sailing under different load conditions, weather conditions, speed, sailing distance, and drafts. Using noon-report data, an R^2 of approximately 90% is obtained through the best performing modelling approaches. However, ADLM systems can increase modelling R^2 by 5 to 7% compared to noon-reports, whilst reducing the required data acquisition period by up to 90%.

Optimising hyperparameters will increase model's R^2 coefficient, but evaluating several modelling architectures should be the first step. ETRs, RFRs, SVRs, and ANNs yielded the best performance results for both datasets, but LR, a significantly simpler model, attained comparable results.

However, due to the inherent limitations of DTR-based models (e.g. ETRs, RFRs), these models should only be preferred in cases where no extrapolation is required. Moreover, the quality of the model output correlates with the quality of its training input; e.g. different vessel operating profiles affect how the effects of vessel draft are perceived to affect FOC. Finally, feature extraction did not help attain any perceivable increase in model performance.

6.3 Vessel FOC prediction & weather routing

This section presents and discusses the results obtained from the vessel FOC prediction and weather routing case study originally described in Section 5.3. The aim of this case study is to (a) ascertain that the FOC of this vessel can be accurately predicted under varying sailing and operational conditions; and (b) demonstrate that the novel weather routing methodology proposed in Section 4.9 can accurately identify optimal sailing

routes for the investigated voyage. This is exhibited through a case study concerning a 160,000 tonne DWT crude oil tanker sailing between the Gulf of Guinea and the Marseille anchorage.

Following the pre-processing steps described in Section 5.3, 647 points are dropped after filtering for anomalies and transients, leaving 2066 points in the dataset.

The descriptive statistics of the dataset are presented in Table 6.11, following the application of the pre-processing and feature engineering process described in Chapter 4, to present the variability of each parameter. This is crucial to ensure the robustness and usability of the produced model.

Table 6.11: Descriptive statistics of the training part of the dataset used in the vessel FOC prediction and weather routing case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	mean	std	min	25%	50%	75%	max
Draft fore	14.14	3.18	5.15	13.93	15.59	15.68	16.72
Draft aft	15.39	2.53	8.07	15.05	16.52	16.77	17.01
Engine speed	59.82	3.28	44.00	59.90	59.90	61.00	68.00
Engine FOC	1253.85	187.35	525.75	1228.88	1284.33	1347.70	1990.61
Engine power	7176.90	998.75	3148.50	6849.59	7278.97	7618.12	12580.75
Vessel over ground speed	11.77	0.99	3.95	11.30	11.80	12.40	14.60
long. current	-0.04	0.24	-1.07	-0.12	-0.02	0.07	0.67
long. primary swell wave significant height	0.02	0.97	-3.35	-0.62	0.07	0.63	3.23
long. secondary swell wave significant height	0.02	0.43	-1.46	-0.23	0.01	0.27	1.56
long. wave significant height	0.07	1.25	-4.19	-0.83	-0.02	0.98	3.89
trans. current	0.17	0.19	0.00	0.05	0.11	0.22	1.15
trans. primary swell wave significant height	0.73	0.54	0.00	0.29	0.63	1.04	3.02
trans. secondary swell wave significant height	0.32	0.29	0.00	0.11	0.24	0.44	1.33
trans. wave significant height	1.01	0.69	0.00	0.49	0.92	1.41	3.95

6.3.1 FOC model derivation

Applying the FOC modelling methodology discussed in Section 4.6.1.7, the optimal ANN hyperparameters are identified as shown in Table 6.12.

Testing the model derived on a test set that corresponds to real vessel sailing data,

Table 6.12: Optimal ANN hyperparameter values for the vessel FOC prediction and weather routing case study. 2,3 or 4 hidden layers of varying width were considered, along with different dropout rates, activation functions, and batch sizes.

Hyperparameter	**Value**
Network depth	4
Layer 1 width	256
Layer 2 width	256
Layer 3 width	64
Layer 4 width	512
Layer 1 Dropout rate	0.36
Layer 2 Dropout rate	0.47
Layer 3 Dropout rate	0.04
Layer 4 Dropout rate	0.53
Layer 1 Activation function	ReLU
Layer 2 Activation function	ReLU
Layer 3 Activation function	Sigmoid
Layer 4 Activation function	ReLU
Batch size	32

an R^2 of 89.4% is obtained. The results obtained are presented in the form of a Y-Y plot in Figure 6.22. Furthermore, the error between predicted and actual FOC for the test set is presented in time-series form in Figure 6.23. In this graph, 95.5% of predicted points are shown to be within 10% of their actual values. A limited number of outliers cross this value and present increased error values but these values are only obtained sporadically and can be attributed to discrepancies between the actual weather conditions observed in the vessel's ambient space and the weather provider's extrapolated data. In practical terms, having a model that can achieve a high R^2 on real vessel sailing data is a practical validation of the model's fitness for this purpose.

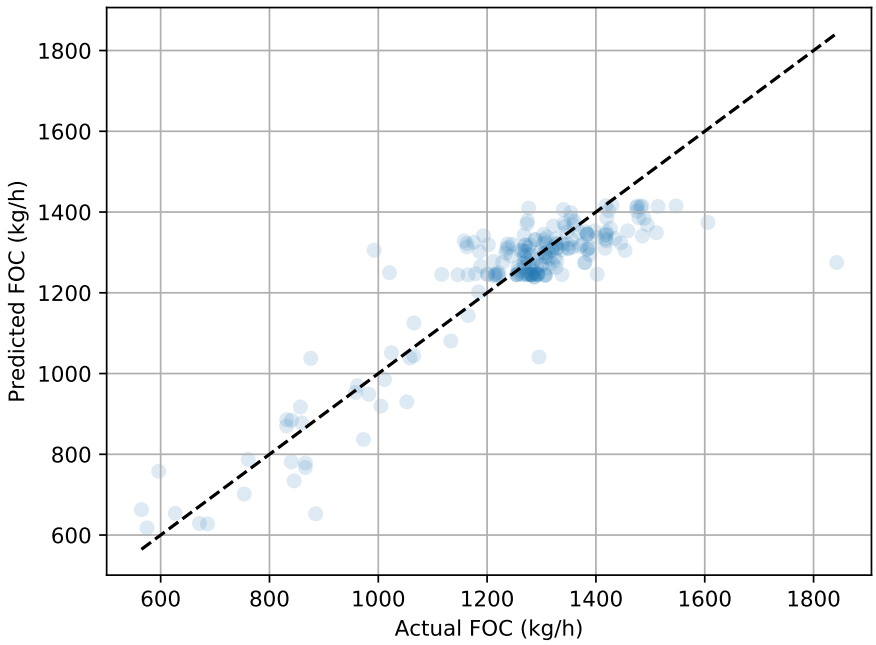

Figure 6.22: Y-Y plot depicting the FOC modelling results in the vessel FOC prediction and weather routing case study. X axis depicts the actual vessel fuel consumption and the Y axis depicts the predicted value. the dashed line is the $y = x$ line, i.e. the line where the predicted FOC value matches the one actually observed.

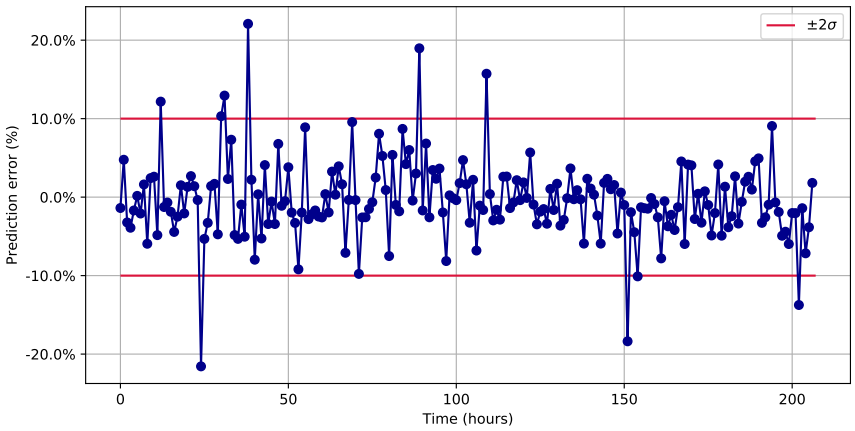

Figure 6.23: Timeseries plot depicting the prediction error over time in the vessel FOC prediction and weather routing case study. Error is presented in percentage form. The red lines denote the $\pm 2\sigma$ range for this graph.

6.3.2 Weather routing

Following the FOC model validation, the routing methodology of Section 4.9 is applied to the route described above. As mentioned in Section 5.4, the methodology is applied for two case studies where the vessel is sailing at two speeds, 11 knots and 14.5 knots. These vessel speeds refer to two typical scenarios of vessel speed whilst fully laden, and in ballast condition. Five optimal routes are identified for each case study, presented in Figures 6.24 and 6.25 accordingly.

In Figure 6.24 where the vessel is sailing at 11 knots, the red (thickest) line corresponds to the route that minimises FOC followed by the yellow, blue, green, and pink lines in diminishing order of thickness, and optimality. In this case, all routes suggested follow the contour of Africa before heading through the strait of Gibraltar and then contour the Iberian peninsula till Marseille anchorage is reached. In this case, two paths that yield similar results can be observed: one opts for a straight-line approach leading to the shortest path whilst the other opts for a longer, lower FOC-per-nm route that accurately follows the gulf that appears between the Republic of Congo, DRC, and Angola, balancing the longer distance by the existence of more favourable weather conditions.

In Figure 6.25 where the case study vessel is sailing at 14.5 knots, with routes similar to those of Figure 6.24 being identified as optimal. However, in this case the contour of Africa is followed less strictly, especially in the area around Mauritania. This is mostly due to weather differences and the vessel's performance changing under varying speeds, a different optimal path is identified. Both of the derived routes are in line with the routes this vessel follows during its normal sailing activities.

In cases where larger route deviations are permitted in order to avoid adverse weather conditions, such as a crossing of the Atlantic, the weather optimisation module would return paths with lower overlap levels.

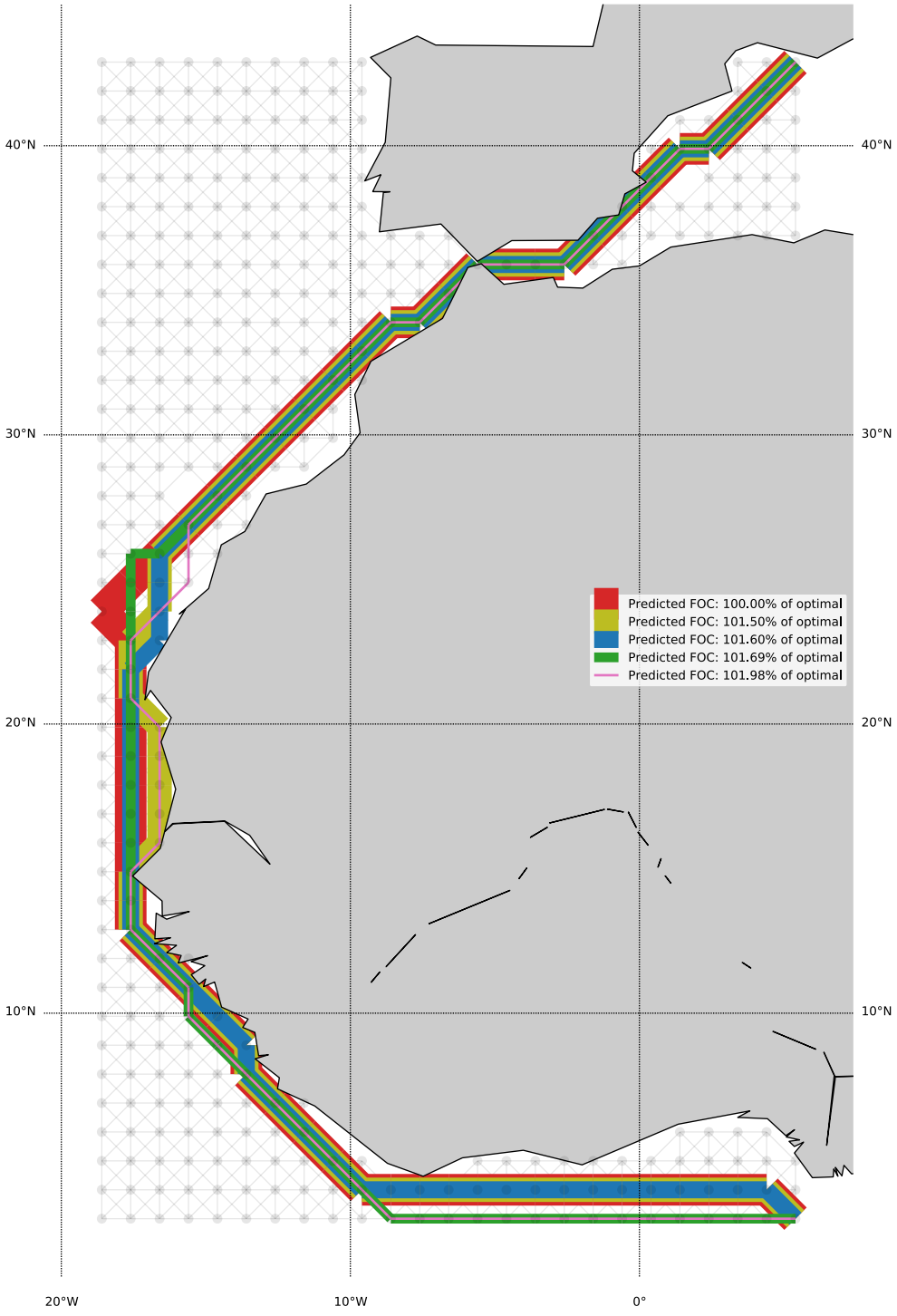

Figure 6.24: Optimal routes identified for the vessel FOC prediction and weather routing case study when the vessel is sailing at 11 knots. Different colours denote different identified routes. The gray grid denotes the graph over which weather routing was applied.

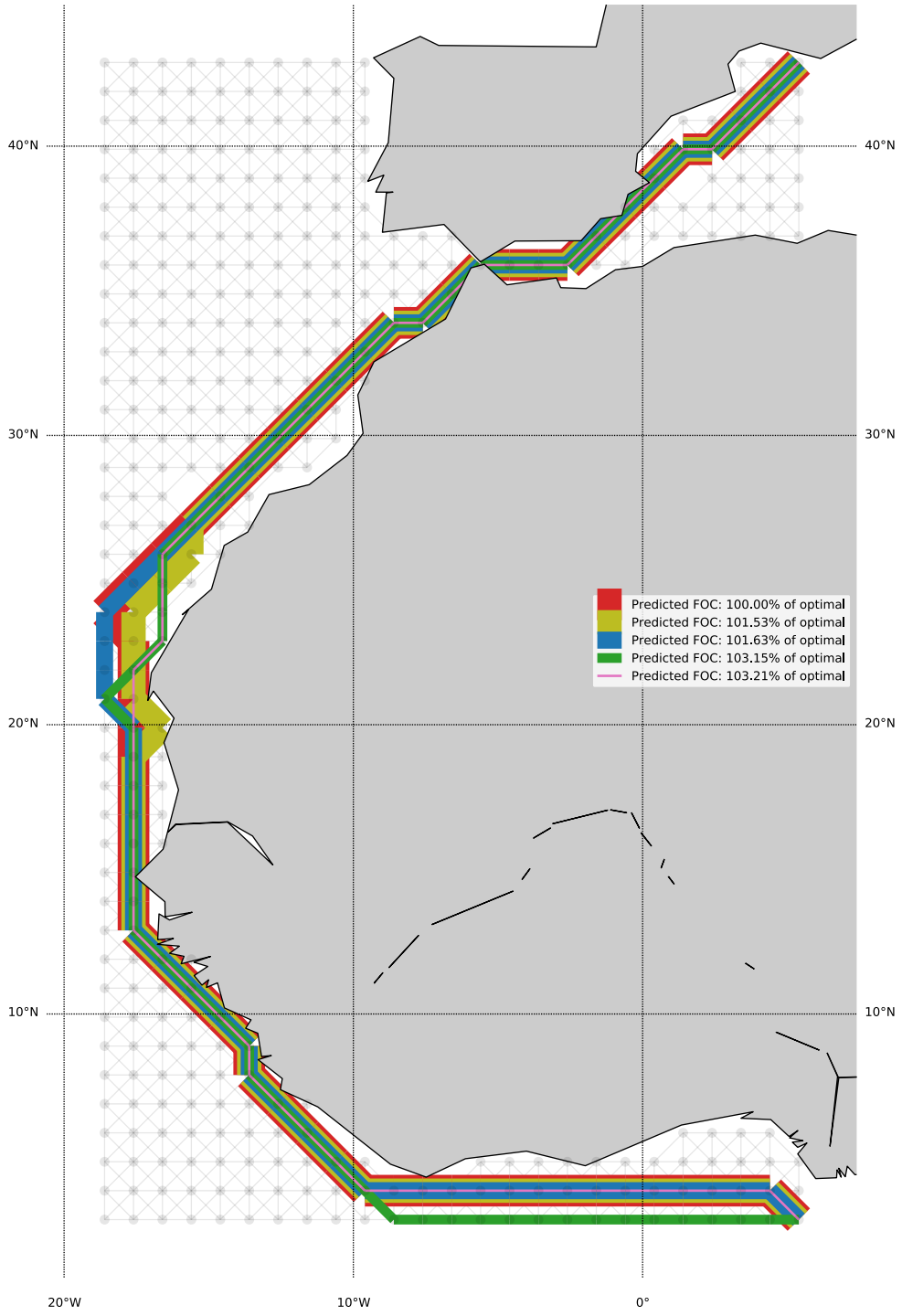

Figure 6.25: Optimal routes identified for the vessel FOC prediction and weather routing case study when the vessel is sailing at 14.5 knots. Different colours denote different identified routes. The gray grid denotes the graph over which weather routing was applied.

6.3.3 Key findings

This Section presents the results of a case study based on the novel weather routing methodology elaborated in Section 4.9, i.e. a data-driven methodology for optimal routing decision support. This methodology is elaborated and showcased through the case study of a 160,000 tonne DWT crude oil tanker sailing between the Gulf of Guinea and the Marseille anchorage.

The initial process of acquiring, pre-processing and analysing a dataset containing raw sailing measurements from a ship is showcased in detail, elaborating on every required step. Moreover, the process to derive a data-driven Deep Neural Network (DNN) model that can adequately represent the FOC of a vessel sailing under varying conditions is then included. The FOC model is utilised as part of a route optimisation process that combines the data-driven aspect of the FOC with a modification of Dijkstra's algorithm to allow for time-dependent route optimisation.

As this constitutes a data-driven, black-box approach to weather routing, the only required data are historical vessel sailing data and a suitably spatio-temporal weather forecast without any additional domain knowledge. Therefore, the proposed methodology can be applied to any route and vessel type without significant modifications.

6.4 FOC-based performance monitoring

This section presents and discusses the results obtained from the FOC-based performance monitoring case study originally described in Section 5.4. This methodology aims to provide vessel performance monitoring using readily available data, including noon reports. The case study is based on data from a a 1100 TEU reefer vessel.

Following the pre-processing steps described in Section 5.4, observations where features with NaN values are removed, with 671 data points remaining. Following this, outliers and observations where less than 10 steam hours are also discarded to avoid observations where mostly transient vessel operation has occurred. The descriptive statistics of this

dataset are presented in Table 6.13.

Table 6.13: Descriptive statistics of the training part of the dataset used in the FOC-based performance monitoring case study. 25, 50, and 75% rows correspond to the relevant percentiles.

	Distance run (nm)	Vessel Speed (kn)	M/E Speed (RPM)	M/E FOC (tn/day)	Sea Current (kn)	Wind Speed (m/s)	Wind Direction (degrees)	Sea Force (m)	Sea Direction (degrees)	Draft forward (m)	Draft aft (m)
count	512.00	512.00	512.00	512.00	512.00	512.00	512.00	512.00	512.00	512.00	512.00
mean	321.92	14.62	109.22	20.03	0.22	4.40	5.34	3.52	5.44	6.51	7.69
σ	67.55	1.65	11.62	6.76	0.73	1.23	3.13	1.21	3.09	1.08	0.61
min	134.00	8.21	62.00	6.30	-4.00	2.00	1.00	1.00	1.00	3.50	6.20
25%	282.50	13.75	98.75	14.50	0.00	3.00	3.00	3.00	2.00	5.88	7.26
50%	341.50	14.85	112.00	20.60	0.00	4.00	5.00	3.00	5.00	6.66	7.70
75%	373.92	15.88	119.00	26.00	0.50	5.00	8.00	4.00	8.00	7.30	8.20
max	433.00	18.00	125.20	32.00	4.00	12.00	12.00	7.00	12.00	8.50	8.94

Based on the information depicted in this table, 512 observations were retained in the dataset following its pre-processing and outlier rejection. Moreover, it can be seen that the vessel speed in this dataset remained relatively stable with about half the observations lying within a knot of 14.8 knots. Observed sea current values were limited, although a maximum value of 4 knots was observed. Furthermore, 95% of the observed drafts were within two meters of the mean draft, in line with the vessel's expected operational profile as a feeder-sized containership.

6.4.1 LR model

An LR model is selected as the first choice for model training as, due to the lack of hyperparameters, is easy and quick to train and provides a solid baseline. An R^2 score of 78.80% is obtained at training.

The model evaluation on data from the seventh month, eighteenth month, and twenty-fifth month post-launch are presented in Figures 6.26, 6.27, and 6.28 respectively. Moreover, Key Performance Indicator (KPI) I of Section 4.8 is calculated and superimposed on each graph. This KPI reflects the vessel's sailing performance, i.e. the ratio between the measured and the expected distance travelled over a given amount of fuel. Other KPIs cannot be calculated for this dataset due to the lack of a power/torque meter.

In the case of the data corresponding to the seventh month, the R^2 obtained is 69.54%

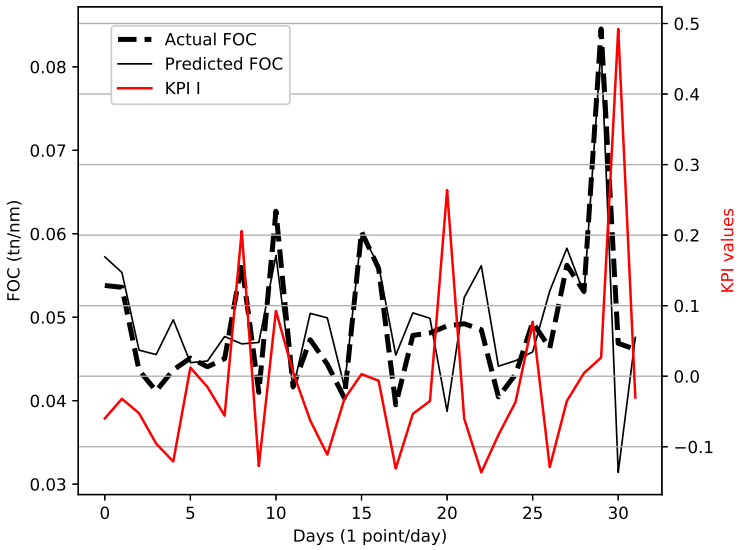

Figure 6.26: Timeseries plot depicting actual against LR-predicted FOC in the FOC-based performance monitoring case study based on data from the seventh month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the LR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

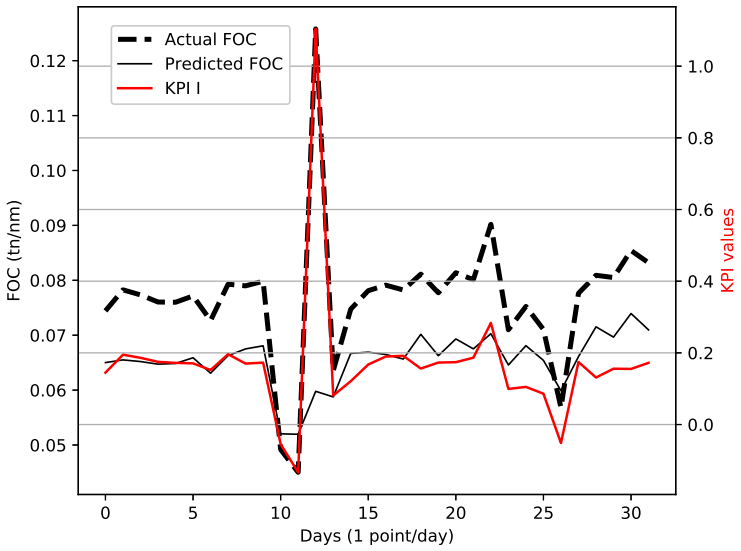

Figure 6.27: Timeseries plot depicting actual against LR-predicted FOC in the FOC-based performance monitoring case study based on data from the eighteenth month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the LR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

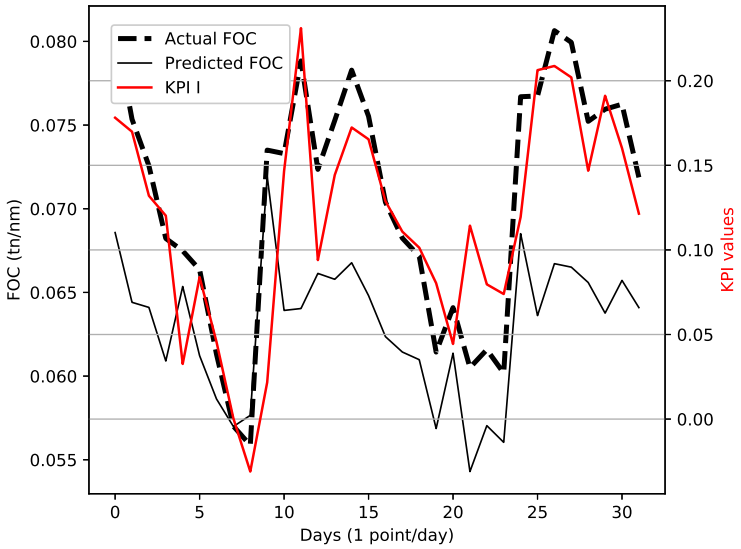

Figure 6.28: Timeseries plot depicting actual against LR-predicted FOC in the FOC-based performance monitoring case study based on data from the twenty-fifth month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the LR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

with a median FOC deviation of 0.0029 tonnes per nautical mile. Evaluating against the data from the eighteenth month, an R^2 of -51.41% is obtained with a median FOC deviation of 0.0103 tonnes per nautical mile. Finally, in the case of the twenty-fifth month, an R^2 of -57.68% is obtained, with a median FOC deviation of 0.0149 tonnes per nautical mile.

6.4.2 SVR model

In the case of the SVR model, results are presented in Figures 6.29, 6.30, and 6.31 respectively, for the results corresponding to the seventh, eighteenth, and twenty-fifth month. An R^2 of 87.00% is obtained at training. Furthermore, evaluating the results of the seventh month, an R^2 of 82.27% is obtained, with a median FOC deviation of 0.0026 tonnes per nautical mile. At the eighteenth month, the R^2 obtained is at -42.05% with a median FOC deviation of 0.0082 tonnes per nautical mile. Finally, at the twenty-fifth month, the obtained R^2 is at -77.00% with a median FOC deviation

of 0.0123 tonnes per nautical mile.

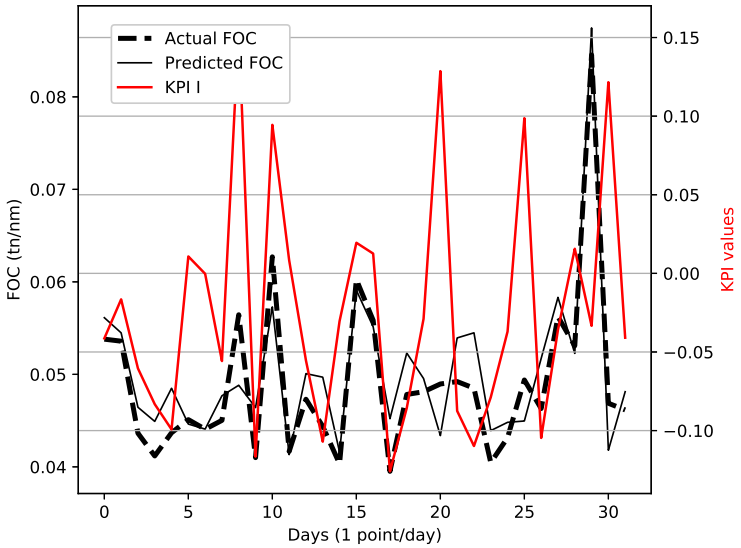

Figure 6.29: Timeseries plot depicting actual against SVR-predicted FOC in the FOC-based performance monitoring case study based on data from the seventh month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the SVR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

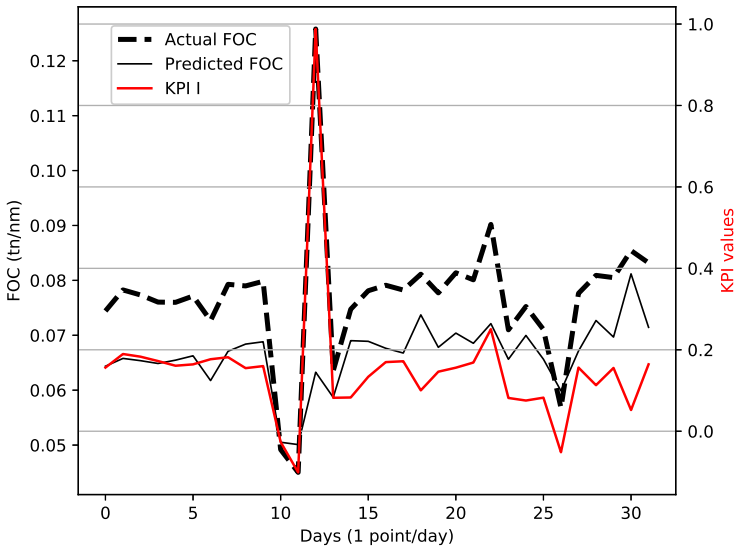

Figure 6.30: Timeseries plot depicting actual against SVR-predicted FOC in the FOC-based performance monitoring case study based on data from the eighteenth month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the SVR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

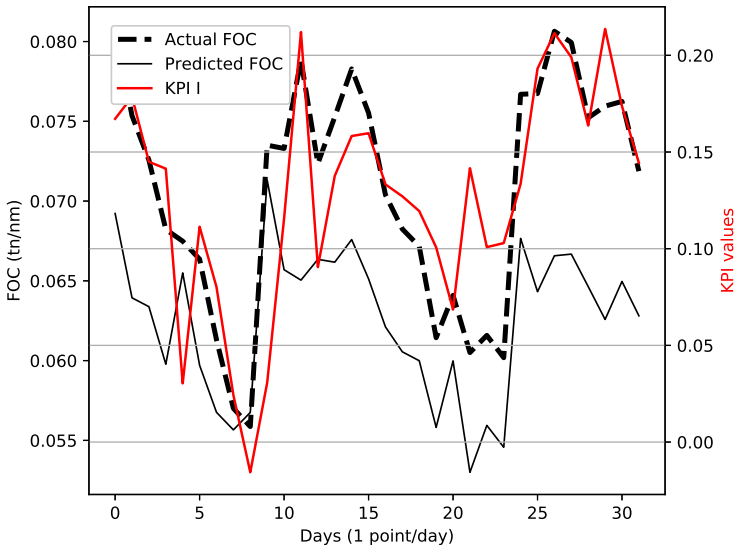

Figure 6.31: Timeseries plot depicting actual against SVR-predicted FOC in the FOC-based performance monitoring case study based on data from the twenty-fifth month following the vessel's launch. The black dashed line represents the actual FOC observed while the solid black line represents the SVR model prediction. The red line displays the calculated value of KPI I, i.e. vessel sailing performance.

6.4.3 Key findings

This Section presents the results of a case study based on the FOC-based performance monitoring methodology of Section 4.8. This methodology, elaborates a data-driven approach for the construction of FOC predictive models using data following a vessel's launch of major overhaul and comparing their predictions with the vessel's actual FOC at later time points to estimate a performance drop. This methodology was elaborated using data from a 1100 TEU reefer vessel.

Based on the results of this case study, both derived models can accurately predict the vessel's FOC when tested against the dataset corresponding to the seventh month post launch. Accuracy is still lower than what is observed in the FOC prediction case studies of Sections 5.2 and 5.3 due to the small amount of training data used. As expected from the results of the case study presented in 5.2, the SVR model performs marginally better than the LR model.

However, evaluating the models against the dataset corresponding to the eighteenth month, performance of both models drops significantly. At the same time, the relevant KPI calculated, increases, denoting a larger degree of dissimilarity between measured and expected FOC. Model performance drops further when the dataset corresponding to the twenty-fifth month is considered. At the same time, KPI values increase even more. This performance degradation was known to the vessel operator and was due to hull fouling. Underwater hull cleaning was performed shortly after the end of the provided dataset.

6.5 Overall discussion

This section summarises the discussion and key findings of the results sections above. Overall, the included case studies demonstrate the applicability and robustness of the proposed methodologies, unified under the aim of enhancing a vessel's operational efficiency. The findings of each case study are summarised below, with each section corresponding to one of the case studies elaborated in this Section.

6.5.1 Engine Condition Monitoring

The case studies relating to engine condition monitoring aim to exhibit the novel characteristics of the condition monitoring methodology presented in Section 4.5. The methodology was applied on one two-stroke main engine and a four-stroke engine used for electrical power generation. The aim of this case study was to evaluate the accuracy of the novel condition monitoring methodology in identifying incipient machinery anomalies.

The main findings of the case studies can be summarised as follows:

- The developed model can accurately discern between normal and abnormal machinery condition, providing a suitable basis for an incipient-fault detection system.
- Due to limited amount of assumptions, this methodology can be applied to a diverse set of machinery.
- Contrary to most other condition monitoring approaches, model training only requires normal data, majorly simplifying the process.
- The data requirements of the proposed methodology can be satisfied from noon-report data, significantly decreasing, or even avoiding, CAPEX costs traditionally attached to condition monitoring applications.
- Using the EWMA filter, the proposed methodology becomes dynamic as previous data points are taken into consideration when predicting the normality of new data points.
- Through the performed sensitivity analysis, model alarm thresholds were obtained. Comparing these values with the ones provided by OEMs, a significant degree of similarity was observed.
- Even in cases were only a limited amount of OEM thresholds are available, relevant threshold can be suggested by analysing model output.

6.5.2 FOC modelling comparison

The two case studies relating to the FOC modelling comparative methodology described in Section 4.6 aim to compare the performance of data-driven regression models in the estimation of a vessel's FOC under different circumstances. The aim of this case study is to identify optimal FOC modelling approaches when different data sources exist, namely noon reports and ADLM systems.

The key findings are summarised below.

- The derived models can accurately predict the FOC of vessels sailing under different load conditions, weather conditions, speed, sailing distance, and drafts.
- Using noon-report data, an R^2 of approximately 90% was obtained through the best performing modelling approaches.
- ADLM systems can increase modelling R^2 by 5 to 7% compared to noon-reports, whilst reducing the required data acquisition period by up to 90%.
- Optimising hyperparameters may increase model's R^2 coefficient, but evaluating several modelling architectures should be the first step.
- ETRs, RFRs, SVRs, and ANNs yielded the best performance results for both datasets, but LR, a significantly simpler model, attained comparable results.
- Due to the inherent limitations of DTR-based models (e.g. ETRs, RFRs), these models should only be preferred in cases where no extrapolation is required.
- The quality of the model output correlates with the quality of its training input; e.g. different vessel operating profiles affect how the effects of vessel draft are perceived to affect FOC.
- Feature extraction did not help attain any perceivable increase in model performance.

6.5.3 Vessel FOC prediction & weather routing

This case study concerned a 160,000 tonne DWT crude oil tanker sailing between the Gulf of Guinea and the Marseille anchorage. This route is selected as it represents a typical voyage of the aforementioned vessel. The aim of this case study was to (a) ascertain that the FOC of this vessel can be accurately predicted under varying sailing and operational conditions; and (b) demonstrate that the novel weather routing methodology proposed in Section 4.9 can accurately identify optimal sailing routes for the investigated voyage.

The main findings of this case study are as follows:

- The initial process of acquiring, pre-processing and analysing a dataset containing raw sailing measurements from a ship is elaborated.
- The process to derive a data-driven DNN model that can adequately represent the FOC of a vessel sailing under varying conditions is then included.
- Moreover, the FOC model is utilised as part of a route optimisation process that combines the data-driven aspect of the FOC with a modification of Dijkstra's algorithm to allow for time-dependent route optimisation.
- The above are verified and validated through a case study based on actual ship data.

6.5.4 FOC-based performance monitoring

This case study stemmed from the FOC-based performance monitoring methodology described in Section 4.8. The underlying methodology aims to provide vessel performance monitoring using readily available data, including noon reports. Specifically, a data-driven FOC predictive model is built based on data obtained just after the launching of the investigated vessel and the model results are compared to the vessel's actual performance in terms of FOC.

The key findings of this case study are as follows:

- Data driven models were trained using data corresponding to a time period just following a vessel's launch.
- Both the LR and the SVR models can accurately predict vessel FOC when tested against data obtained close to the time period of the training dataset.
- The SVR model performed marginally better.
- As model testing is performed using data acquired months after the acquisition of the training data, model accuracy drops significantly.
- At the same time, the values of the calculated KPI that reflects vessel sailing performance are increased, denoted degraded performance.
- Therefore, vessel voyage data can be used as input to a FOC predictive model for performance monitoring purposes.

6.6 Chapter summary

This chapter presented the results of the case studies considered for the evaluation and validation of the operational efficiency enhancement framework proposed in Chapter 4. This framework comprises four key methodologies, namely: the condition monitoring methodology; the FOC modelling comparative methodology; the weather routing methodology; and the FOC-based performance monitoring methodology. Accordingly, the case studies elaborated in Chapter 5 reflect these key methodologies and are are designed to exhibit the capabilities of the underlying framework in realistic operating conditions. For this reason, the results of these case studies are presented in this chapter. First the results of the engine condition monitoring case study were presented, followed by the case study demonstrating the results of the FOC modelling comparison case study. These were followed by the results of the vessel FOC prediction and weather routing case study. Finally, the results of the FOC-based performance monitoring case study were presented, followed by a Section that provides an overall discussion of the results obtained.

Chapter 7

Discussion & Conclusions

This Chapter contains the discussion and concluding remarks regarding the operational efficiency enhancement framework presented and developed in Chapter 4. The research aim and objectives originally laid out in Chapter 2 are reviewed, and the manner they were accomplished is analysed. Following that, the novelty of the research contained within this thesis is summarised and discussed. Furthermore, conclusions to the overall work presented within this thesis are presented, followed by recommendations for future research.

7.1 Accomplishment of research aim and objectives

The main purpose of this research is to practically and theoretically contribute to the enhancement of operational efficiency in the maritime field. This was achieved by addressing the research question posed in Chapter 2, regarding the development of a data-driven strategy for machinery condition monitoring and identification of incipient anomalies, vessel performance degradation monitoring, and routing decision support through the proposed methodologies. In this respect, this main aim was addressed overall through the thesis objectives outlined in Chapter 2. These objectives and the way they were addressed are discussed in more depth in this section.

Objective 1: Identify gaps pertinent to the research topic through the critical review of literature pertinent to data-driven monitoring, optimised maintenance and routing.

This objective has been achieved through the critical review of maintenance strategies, condition monitoring applications, data-driven Fuel Oil Consumption (FOC) modelling applications, and weather routing approaches that is part of Chapter 3. Critical aspects, advantages and shortcomings of relevant methodologies have been identified, defining the path towards novel research and development activities aiming to enhance vessel operational efficiency. As part of this mapping, solutions stemming from both the maritime and other relevant sectors were identified and analysed. This review highlighted that the maritime sector is often lagging compared to other sectors. However, this review also showed that often there no clear path forward, and that instead, approaches need to be evaluated on a case-by-case basis, ensuring the robustness of each step. Moreover, the scarcity of integrative approaches was demonstrated, as many of the identified frameworks and methodologies address a single objective, without being integrable with other tools to provide an applicable solution towards operational efficiency enhancement.

Objective 2: Consider and address the identified gaps through the development of a streamlined methodology that enhances vessel operational efficiency.

This objective has been achieved through the proposal and discussion of the overall research framework, proposed in Chapter 4, and visually depicted in Figure 4.1. The overall framework comprises a number of key methodologies, namely: the data preprocessing methodology, the condition monitoring methodology, the FOC modelling comparative methodology, the FOC-based performance monitoring methodology, and the novel weather routing methodology. This framework enables the anomaly detection in marine machinery systems, the identification of optimal FOC modelling architectures, FOC-based performance monitoring, and optimal routing decision support. All these outcomes, which have never before been considered under a single framework within the maritime sector, are summarised under the overall aim of this thesis,

i.e. that of vessel operational efficiency enhancement. The flexibility and robustness of the framework and methods proposed within the thesis, allow their application in a variety of scenarios, relevant both within and outwith the shipping sector.

Objective 3: Extract meaningful information from available data sources through the development of a suitable data pre-processing methodology.

This objective has been achieved through the elaboration of a suitable data pre-processing methodology in Section 4.3. This methodology comprises five salient aspects: engine transients detection, recording anomalies rejection, weather data imputation, feature engineering, and data standardisation. Process, voyage, and weather data are treated, with different steps applied to each. The output of this methodology is a cleaned-up, accurate stream of data that is fed to the remaining methodologies of this research work.

Objective 4: Monitor the condition of machinery and detect incipient anomalies through the development of a suitable data-driven methodology, limiting the number of data-related assumptions.

This objective has been achieved through the novel data-driven Condition Monitoring (CM) methodology proposed in Section 4.5. This methodology takes as input pre-processed data from a machinery system and uses them to train a One Class Classification (OCC) model. This model can then evaluate newly acquired data points and detect machinery system anomalies. The novel aspect of this methodology is that it constitutes the first time a OCC model is used for anomaly detection in a shipping context, without any requirement for faulty data. These simplified data requirements allows the rapid deployment of the model, without requiring that a fault occurs so that data can be recorded prior to deployment.

Objective 5: Identify optimal data-driven modelling architectures for the prediction of vessel FOC through a formalised, novel methodology.

This objective has been achieved through the novel FOC modelling comparative meth-

odology proposed in Section 4.6. This methodology considers data sourced from two different approaches, namely noon reports and Automated Data Logging & Monitoring (ADLM) system data, and a number of regression modelling architectures and identifies best model architectures in each case. Furthermore, the advantages of having data acquired through an ADLM system, are quantified in terms of increased model accuracy. The novel aspect of this methodology is that it compares both data sources through the same formalised modelling pipeline, allowing the observation of meaningful conclusions. This methodology allows downstream methodologies that process FOC data to identify optimal modelling architectures, increasing their accuracy and robustness.

Objective 6: Monitor the performance degradation of vessels based on FOC modelling through the development of a suitable data-driven methodology.

This objective has been achieved through the FOC-based performance monitoring methodology elaborated in Section 4.8. This methodology trains a data-driven regression model using voyage data acquired following a vessel's launch of major overhaul aiming to predict its FOC. Given that the model is trained using data reflecting optimal vessel condition, the FOC predictions will inherently contain the same assumption. Therefore, comparing the predicted FOC with the one actually observed on a voyage, allows the estimation of performance degradation through the increase of FOC. This methodology allows the provision of vessel performance degradation monitoring using readily available data, and without the need of explicit vessel-dependent monitoring.

Objective 7: Facilitate optimal routing through the development of a suitable Decision Support methodology.

This objective has been achieved through the optimal routing decision support methodology proposed in Section 4.9. This methodology considers historical weather and voyage data to train a data-driven regression model that predicts vessel FOC under varying sailing and ambient weather conditions. This is coupled with a novel modification of Dijkstra's algorithm to make it dynamic, i.e. consider weather changes at

different time steps. This methodology allows the identification of overall optimal sailing paths, and therefore offer optimal routing decision support.

Objective 8: Demonstrate and validate the performance and applicability of the framework and methodologies developed through case studies reflecting realistic scenarios applicable to a variety of vessel types (e.g. reefer, containership, bulk carrier).

This objective has been achieved through the case studies presented and discussed in Chapters 5 and 6. Case studies relevant to engine condition monitoring, the FOC modelling comparative methodology, FOC-based performance monitoring, and vessel FOC prediction and weather routing are presented. These case studies showcase and validate all key aspects of the overall research framework, and underlying methodologies summarised in Figure 4.1. Moreover, these case studies present the methodology's capabilities in addressing its main targets, namely: anomaly detection, FOC-based performance monitoring, and optimal routing decision support; all contributing to the overarching aim of enhancing vessel operational efficiency. The contribution of each case study in meeting this eighth objective are summarised below.

The engine condition monitoring case study considered three realistic engine scenarios, all corresponding to a 439 TEU Reefer. Two scenarios relate to the operation of a four-stroke Diesel Generator set (D/Gen) at different time points and one relates to the operation of a two-stroke Main Engine (M/E). Training the OCC models using data corresponding to nominal system operation, sensitivity analysis was performed, evaluating model output in varying input data. In all cases, the developed models accurately discerned between normal and abnormal system states, with the model identifying anomalies at values close to the limits provided by the Original Equipment Manufacturer (OEM).

The FOC modelling comparison case study considered a 1100 TEU Reefer vessel where data were obtained through noon reports, and a Newcastlemax Bulk carrier equipped with an ADLM system. For both vessel datasets, the same number of data points were selected and used through the same model training pipeline, evaluating a large

number of data-driven regression models in order to determine which model architecture performs best in each case. Support Vector Reggressor (SVR) and Artificial Neural Network (ANN) architectures provided the best results in both cases, with an R^2 of approximately 90% obtained in the case of the dataset considering noon reports. In the case of the ADLM system, R^2 was increased by over 5% whilst reducing the required data acquisition period by up to 90%.

The FOC-based performance monitoring case study relates to the FOC modelling comparative methodology as it also relates to data-driven regression modelling for FOC prediction. However, in this case, measurements from the first six months post launch from a 1100 TEU reefer were used to train a regression model predicting its FOC. When the model was evaluated against data obtained close to the training dataset, a high level of model accuracy was observed. However, as the model was evaluated against data corresponding to one year later, and one-and-a-half year later, significant deviations between predicted and observed FOC was determined. Thus, this case study validated the proposed case study as an efficient approach to vessel performance monitoring using readily available data.

Finally, the vessel FOC prediction and weather routing case study evaluated the proposed weather routing methodology through the consideration of a Gulf of Guinea to Marseille anchorage voyage of a 160,000 tonne DWT crude oil tanker. This case study showcased that when a FOC regression model is trained using a diverse dataset, a high model accuracy can be observed in varying operating conditions. Furthermore, optimal weather routes were obtained for two vessel speeds considered, reflecting its operation when laden and at ballast. Therefore, this case study showcased the methodology's capability in delivering optimal vessel routes, with the validity of the results being validated by the consultancy company that supported this research.

Summarising, the aforementioned case studies have presented the proposed framework's accuracy and robustness in providing anomaly detection, FOC-based performance monitoring, and optimal routing decision support, therefore enhancing a vessel's operation

efficiency.

7.2 Novelty of presented research

The novelty of the research presented and discussed within this thesis is derived by the development and combination of a number of data-driven methodologies for the operational efficiency enhancement of a vessel. Under the overarching aim of operational efficiency enhancement, three distinct outcomes are provided: anomaly detection, FOC-based performance monitoring, and optimal routing decision support. The proposed and developed operation efficiency enhancement framework introduces novelties in the fields of machinery condition monitoring, vessel performance monitoring, and optimal routing decision support in the maritime industry. Thanks to the data-driven nature of the developed methodologies, these can be applied to a wide number of applications, and integrated with a number of ancillary methodologies that already exist or will be developed in the future. Furthermore, due to the methodologies' flexibility in using ADLM system or noon report input data, as well as using any number of available input sources, the methodologies are extremely adaptable to specific vessel considerations and can be implemented without requiring the installation of specialised sensors, or an *a priori* knowledge of the inner workings of the examined systems.

A novel aspect of this research is the use of a data-driven OCC model for ship machinery condition monitoring, without any requirement for abnormal data availability. The developed methodology takes as input a dataset corresponding to nominal system operation and trains an OCC Support Vector Machine (SVM) model based on that. The trained OCC SVM model can take as input new, unclassified data points and predict whether they correspond to normal or abnormal operation, thus providing incipient machinery anomaly detection at a system level. Concurrently, the results of the applied algorithm are filtered through an Exponential Weighted Moving Average (EWMA) filter, making model output dynamic, as the output takes into account previous time points as well as the present. Moreover, model output can be used to propose operational limits in cases where OEM limits are not available. Furthermore, the sim-

plified data requirements of the model, allow its rapid deployment, without requiring that a fault occurs so that data can be recorded prior to deployment. The characteristics of this methodology make it a beneficial tool to be considered by shipping sector stakeholders, such as ship owners, operators, and crew. Specifically, given that the model can be trained through easily obtainable data and without any other assumptions or requirements, it can provide a low cost, low implementation complexity, early warning system, reducing machinery downtime and increasing a vessel's reliability and availability.

Moreover, another novel aspect of this research is the FOC modelling comparative methodology. This methodology considers two data sources, namely noon reports and ADLM system data, and a number of regression modelling architectures and identifies best model architectures in each case. While data-driven FOC predictive modelling is an emerging research topic, the proposed methodology constitutes the first time that both possible data sources are compared at the same time. This allows the evaluation of both sources in real terms, highlighting the accuracy gains due to the installation of an ADLM system. Thus, this methodology provides actionable information pertinent to any data-driven approach that includes FOC modelling as part of its modelling, including the weather routing and FOC-based performance monitoring methodologies presented within this thesis. This also makes this methodology particularly useful to researchers and developers providing data analytics services to relevant stakeholders.

Accordingly, the proposed weather routing methodology constitutes a novel part of this research work. This methodology includes a heuristic modification of Dijkstra's algorithm, making it dynamic so that changing weather patterns as a function of time. Furthermore, due to the implementation of suitable modifications in this interpretation of Dijkstra's algorithm, optimisation is performed in terms of minimising overall FOC instead of minimising the route distance, as it the case with the traditional implementation of Dijkstra's algorithm. Therefore, this methodology provides a purely data-driven solution to the problem of finding optimal weather. As such, this method-

ology can provide valuable information to operators and crew in terms of reducing the vessel's FOC, and accordingly the vessel's emissions – a highly coveted achievement, both for regulatory bodies and the general public.

7.3 Conclusions

The concluding remarks of this research work are presented in the following statements:

- The presentation of the data-driven modelling framework for vessel operational efficiency enhancement has been enabled through the thorough examination of pertinent literature and research trends from both the maritime sector and other industry sectors with a stronger tradition in data-driven methods. The need for approaches that integrate individual methodologies and tools to provide a more holistic approach to vessel operational efficiency enhancement solidifies the need for a framework such as the one introduced within this thesis.

- Along these lines, the proposed data-driven framework has addressed aspects of machinery condition monitoring, vessel performance monitoring, and optimal routing decision support. Through the combination of a multitude of data-driven approaches, a dynamic framework with proven accuracy and robustness is created, with methodologies that can both used individually and combined for the overarching aim of vessel operational efficiency enhancement.

- The data pre-processing methodology constitutes the first step of the overall operational efficiency framework. This methodology incorporates engine transients and recording anomalies rejection, weather data imputation, feature engineering, and data standardisation. The aforementioned steps work in tandem to pre-process different datasets with different needs and provide data in the form required as input by all downstream methodologies. Combining the remarks of the critical literature review with observations made through the case studies included within this thesis, lack of proper pre-processing can significantly taint datasets,

whilst most industrially-acquired datasets need to be thoroughly cleaned. This illustrates the criticality of an often overlooked part of data-driven methodologies, that of data pre-processing. This proves to be an actionable insight to relevant stakeholders, such as shipowners and operators, as it highlights the need to safeguard data accuracy, starting at the data acquisition step.

- The proposed condition monitoring methodology based on the OCC paradigm provides the benefit of only requiring nominal data to be available at model training. This is contrasting to other data-driven condition monitoring methods that employ the traditional classification paradigm. Through the case studies applying this methodology on data from four-stroke D/Gen and two-stroke M/E, it was shown that the model can accurately discern between normal and abnormal machinery conditions. Therefore, it provides a suitable basis for incipient fault detection systems. Furthermore, due to the minimal amount of latent data assumptions and low data volume requirements, this methodology can be applied to a diverse set of machinery systems. These methodology particularities make it a beneficial tool to be considered by shipping sector stakeholders, such as ship owners, operators, and crew. However, an inherent shortcoming of this methodology is that it can only provide anomaly detection at a system level, without being able to localise anomalies to specific subsystems of components.

- The proposed FOC modelling comparative methodology evaluates different data sources and data-driven regression architectures for the identification of optimal modelling approaches. This methodology and the relevant case study, confirmed that highly accurate FOC predictions can be made even when noon report data is used. Using noon report data, an accuracy of approximately 90% was obtained, while the use of data sourced from an ADLM system yielded an accuracy of approximately 97%. Moreover, this comparative study showed that the use of ADLM systems allows the data acquisition period for model training to be reduced by up to 90%. These observations provide valuable information to key stakeholders such as researchers and developers as they provide guidance on the modelling

approaches that should be prioritised when solving similar modelling problems. In the case of stakeholders such as shipowners and operators, it provides significant insights that can be incorporated into an accurate cost-benefit analysis of the installation of an ADLM system.

- The FOC-based performance monitoring methodology, incorporating aspects of the FOC prediction methodology, elaborates a data-driven approach to monitoring a vessel's performance. This methodology uses FOC as a proxy for performance, identifying instances of unjustifiably increased FOC and flagging them as of degraded performance. This is done using a data-driven regression model that predicts FOC based on data corresponding to a time period of optimal performance, e.g. post the vessel's launch or following a major overhaul. The comparison of model FOC predictions with the value actually observed provides a quantification of the vessel performance degradation. This provides a theoretically sound and easy-to-implement approach, with the results corroborated through a case study using actual ship data. A number of Key Performance Indicators (KPIs) have been proposed as part of this methodology, allowing the translation of the performance degradation into easily understandable, qualitative information. This is particularly useful in the case of stakeholders such as shipowners and operators at it provides useful insights relevant to planning of underwater cleanings, and other maintenance activities.

- The weather routing methodology provide optimal routing decision support aiming to minimise the FOC of a vessel route between two given points. This is achieved through a combination of the FOC prediction methodology and a modification of Dijkstra's algorithm. This novel modification allows the algorithm to acquire dynamic elements, taking into consideration time-dependent weather changes and the derivation of optimal paths based on the minimisation of FOC instead of overall distance minimisation. This weather routing methodology was showcased through a case study utilising actual vessel raw data, and verified by the consultancy company that supported this research work. Based on the above

remarks, This methodology can provide key insights to operators and crew in terms of reducing the vessel's FOC. A corollary of reduced FOC is reduced vessel emissions, therefore satisfying ever more stringent requirements set by regulatory bodies and the general public.

7.4 Recommendation for future research

Through the proposal and development of the operational efficiency enhancement framework contained within this thesis, future research areas that can extend the thesis impact and research scope were identified. These are summarised below:

- Comparison of the OCC SVM model with other emerging OCC classifiers should be considered to improve the condition monitoring methodology. As shown in the FOC modelling comparison case study results, different models can significantly increase model accuracy and robustness. Moreover, the derivation of the algorithmically generated abnormal points for the hyperparameter optimisation of the OCC model could be improved. Besides, the evaluation of the proposed CM methodology using a dataset containing actual faults should be considered to solidify the model's accuracy and usefulness.

- Evolve the proposed CM methodology through the integration of first-principle modelling. This hybrid approach is expected to increase model accuracy, and provide fault localisation and advanced diagnostic capabilities.

- The CM methodology could be extended to obtain forecasting capabilities. This would allow the estimation of future machinery system states, and the provision of early warning in the case of anomalies that may develop to an extent that affects a vessel's operability.

- The overall proposed framework could be extended to become part of an autonomous shipping operations framework. While all aspects proposed within this thesis provide substantial benefits on their own, the actionable information they provide

will become even more important as autonomously operating vessels become more prolific. This could point towards the integration of the proposed framework with tools such as collision avoidance for the provision of a routing framework that considers both global and local constraints.

- The FOC prediction module that is part of the optimal routing decision support methodology could be enhanced by considering the M/E condition and vessel performance estimations, as provided by the CM and FOC-based condition monitoring methodologies. This would improve model accuracy even further, and accordingly provide optimal routing support that is bespoke to the vessel at each given time point.

- The implementation of the CM methodology to other systems based on their criticality and a cost-benefit analysis could be considered. This could be obtained through the integration of Failure Mode and Effect Analysis (FMEA) or Failure Mode, Effects and Criticality Analysis (FMECA) approaches with the proposed data-driven methodology. Furthermore, by examining the underlying causes identifying through this analysis, this methodology could be applied in line with proactive maintenance requirements.

- The impact of a higher volume data source, e.g. an ADLM system should be considered for an additional case study based on the FOC-based performance monitoring methodology presented in Section 4.8. Given a higher volume data source, more advanced model architectures could be considered to improve model accuracy and, accordingly, degradation monitoring. If this is coupled with additional observed parameters (e.g. power/torque measurements), more complex KPIs could be developed to better quantify the performance degradation.

- Optimal routing decision support and vessel performance degradation monitoring through FOC modelling could be improved through the combination of measurements from multiple vessels. This approach would allow the FOC prediction for vessels where a large amount of training data are not readily available, whilst

also providing monitoring at a fleet level. This could be achieved through an implementation of big data principles, including Cloud and Edge computing and the use of Internet of Things (IOT).

- The CM methodology can be integrated with the optimal routing decision support methodology to provide maintenance scheduling optimisation. These methodologies will work in tandem, to identify when a piece of machinery is bound to fail and predict where the vessel will be located at that time to route spares and plan maintenance accordingly.

- The optimal routing decision support methodology can be augmented by considering variable vessel speeds along with weather information. This could be achieved through the implementation of more advanced path-finding algorithms, currently used in other sectors. Moreover, the Machine Learning (ML) paradigm of reinforcement learning could be used, as it is exhibiting promising results in a variety of relevant scenarios, including autonomous navigation.

Appendix A

Taxonomy of data-driven condition monitoring model algorithms

Depending on the goal at hand, data-driven models can been separated in three main groups based on how these models "learn": (i) supervised learning; (ii) unsupervised learning; and (iii) reinforcement learning.

This classification reflects different requirements regarding the use of labelled data. Specifically, in the case of unsupervised learning, patterns in the dataset are identified without the explicit existence of some ground truth. On the other hand, supervised learning requires that every observation in the training dataset has an explicit target (i.e. ground truth) value assigned. A sub-category named semi-supervised learning is often considered to be an extension of supervised learning, combining labelled and unlabelled data for model training. The final model category refers to reinforcement learning where, whilst no labelled data is available, the end goal is explicitly described (often in the form of a cost function), and the algorithm aims to identify the approach that minimises this cost function. Reinforcement learning, although promising in many

applications, does not currently have clear applications in condition monitoring, and therefore will not be considered for the rest of the chapter.

A.1 Unsupervised learning

Unsupervised learning aims to identify some hidden pattern in an unlabelled dataset; this has either the form of dimensionality reduction, or clustering. Dimensionality reduction refers to the replacement of the original dataset features with a new set of features that can describe most of the variance contained in the original dataset whilst reducing the number of required features, therefore helping combat the "curse of dimensionality". "Curse of dimensionality" refers to various undesired phenomena that appear when analysing data in high-dimensional spaces (Bellman, 2003). On the other hand, clustering (often called cluster analysis) aims to group observations within a dataset in different groups based on their similarities.

A breakdown of the most-frequently used algorithms are presented in Figure A.1.

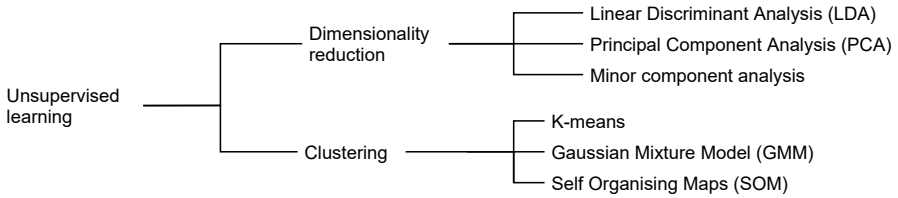

Figure A.1: Taxonomy of unsupervised Machine Learning (ML) algorithms used in Condition Monitoring (CM) applications.

A.1.1 Dimensionality reduction

Dimensionality reduction aims to reduce the number of features contained within a dataset (i.e. the dimensionality of the dataset) while retaining most of the variance (i.e. information) contained within the original dataset. An example of dimensionality reduction is presented and discussed in Figure A.2.

Reducing the dimensionality of a dataset facilitates data storage and enables accelerated data analysis as the sheer amount of data decreases. Burges (2010) notes that the

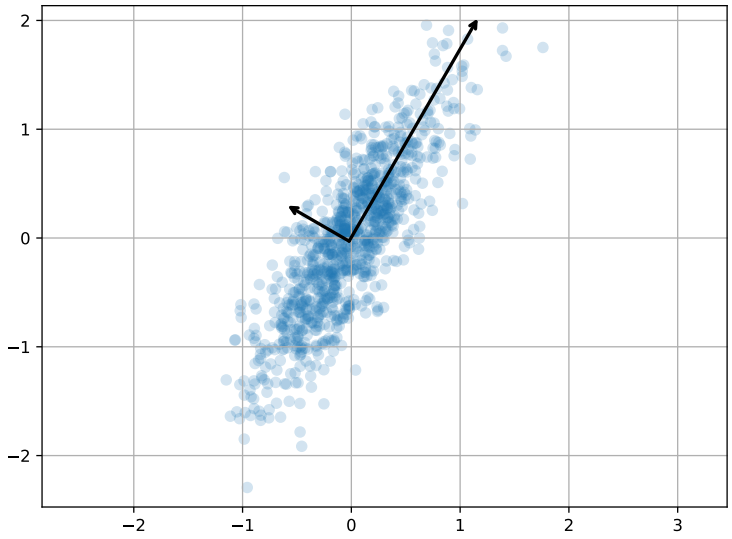

Figure A.2: illustrative example of dimensionality reduction through the application of Principal Component Analysis (PCA) algorithm. The length of each principal component vector represents the amount of variance that is described within that principal component. Thus, it is evident that in this example, keeping only the first principal component (pointing up and to the right) as a dataset feature would still retain most of the relevant information.

optimal rate of regression convergence of a dataset with 20 features and 10 million samples can be replicated with only 10,000 samples if features are reduced to 10 in such a way that the amount of variance contained is not affected. At the same time, as the end features are uncorrelated, the application of such an algorithm may increase the accuracy and robustness of a ML algorithm. Finally, reducing the dimensionality of a dataset facilitates its visualisation. Relevant algorithms found within CM literature are presented below.

Linear Discriminant Analysis (LDA) is a method that identifies a linear combination of features that distinguishes a dataset, first proposed by Fisher (1936). In this respect, LDA can be either used as a dimensionality reduction tool, or as a crude linear classifier (Rao, 1973).

Principal Component Analysis (PCA) is another method used for dimensionality reduction that applies an orthogonal transformation that transform potentially

correlated features to linearly uncorrelated (orthogonal) features called principal components (Hotelling, 1936; Pearson, 1901).

Minor Component Analysis (MCA) is a counterpart to PCA, where the aim is to extract the minor components of a dataset, i.e. the eigenvectors corresponding to the smallest eigenvalues of the autocorrelation matrix whereas PCA aims to identify the eigenvectors corresponding to the largest eigenvalues (Luo et al., 1997). MCA has been proved to be more versatile than PCA in a number of applications (L. Xu et al., 1992).

A.1.2 Clustering

Clustering can be defined as (Hastie et al., 2009):

> [aiming to] partition the observations into groups ("clusters") so that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters.

An example of clustering is depicted in Figure A.3. There, given a dataset and the number of end clusters, the dataset points are assigned into clusters based on the minimisation of their dissimilarities. Some of the most often cited clustering algorithms in CM applications are presented below.

K-means is a very popular (Hastie et al., 2009) clustering algorithm that is intended for situations where all dataset features are quantitative, working iteratively. Observations are assigned to the cluster with the nearest mean.

Gaussian Mixture Model (GMM) is an iterative method that assigns observations to clusters depending on the probability of an observation belonging to each cluster as calculated by a probability distribution. The method iteratively aims to maximise the overall probability of all observations belonging to the respective clusters they have been assigned to. K-means method operates in a similar way, except that is returns a "hard" decision where each observation only belongs to one cluster whereas GMM

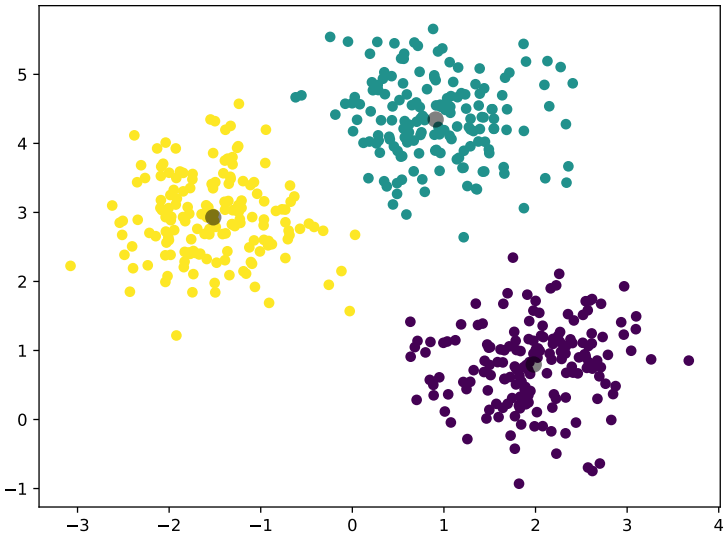

Figure A.3: illustrative example of clustering through the application of k-means algorithm. Given a dataset and the explicit aim to identify 3 clusters, the algorithm returns a dataset clustering so that the dissimilarity between points belonging to the same cluster is minimised.

calculates the probability of a model belonging to all possible clusters. Routinely, the Expectation Maximisation (EM) algorithm is used to obtain the required probability estimates.

Self-Organising Map (SOM) is a type of Artificial Neural Network (ANN) that can be used to produce a lower dimension representation (i.e. map) of the input space (Kaski & Kohonen, 1996). Contrary to most ANN applications, SOM apply competitive learning. While, SOM applications are usually in dimensionality reduction, in the case of the identified CM literature, these applications all use SOM models for clustering purposes.

A.2 Supervised learning

The goal of supervised learning is to predict an output value based on a number of input values (Hastie et al., 2009). If the output value is quantitative the task is called regression, whereas if the output value is qualitative, the task is classification. In both

cases, the model is trained based on past observations where both inputs and outputs are known. A breakdown of the most-frequently used supervised algorithms are presented in Figure A.4. Some of the algorithms presented can be used for both classification and regression tasks, whereas others can only be applied to either classification or regression problems. A special case of classification is one-class classification where all data points available during model training are assumed to belong to one class and new observations are classified as either belonging to that class or not.

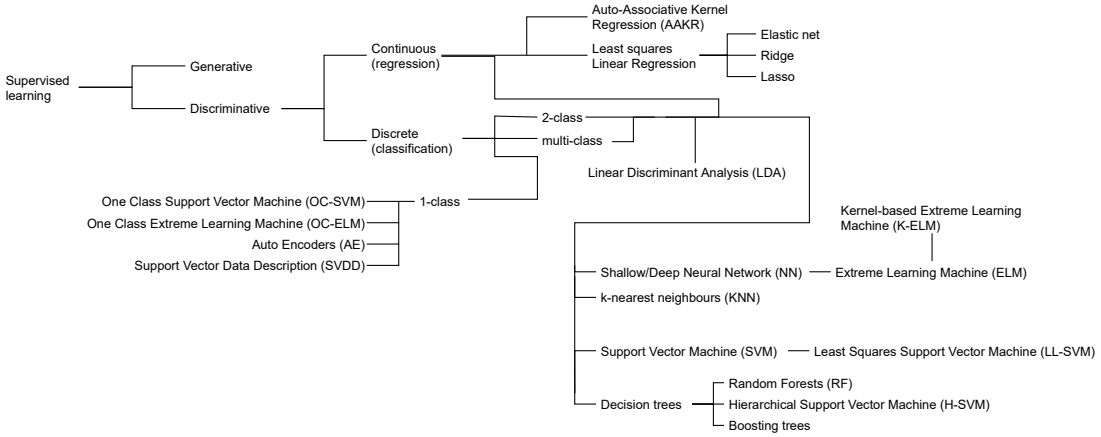

Figure A.4: Taxonomy of supervised ML algorithms used in CM applications.

A.2.1 Regression

Regression aims to predict one or more quantitative values given a number of inputs. An example of regression is depicted in Figure A.5. There, given a number of input and output pairs, a line that connects the two is calculated, providing a prediction of output values for previously unseen input values.

Linear Regression (LR) constitutes the simplest regression algorithm, involving a linear combination of the input (independent) variable (Bishop, 2006). The extension of LR in cases where more than one dataset feature is used as an input is called multiple LR (Hastie et al., 2009). Model parameters are usually computed aiming to minimise the sum of the squares between the observed and predicted dependent (target) variable; a method called least squares.

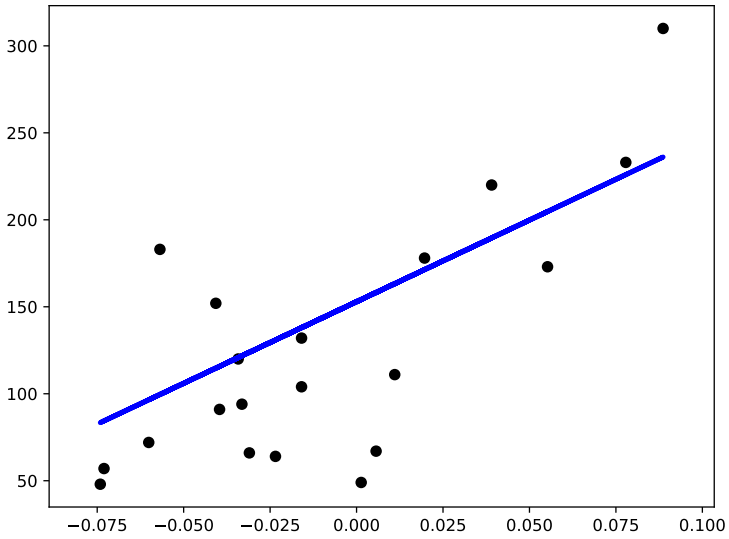

Figure A.5: illustrative example of linear regression. Given a number of input and output pairs, a line that connects the two is calculated, providing a prediction of output values for previously unseen input values.

Ridge Regression (RR) is an extension of LR where the cost function that is to be minimised is similar to that of LR with an additional parameter that imposes a penalty to the square of each parameter. (Hastie et al., 2009). This shrinkage helps avoid overfitting the training dataset. RR is also known as the Tikhonov regularisation.

Least Absolute Shrinkage and Selection Operator (LASSO) is similar to RR but the cost function term imposes a penalty on the absolute value of each parameter instead of their squares.

Elastic net is a combination of RR and LASSO, where both absolute-value and squared regularisations are implemented concurrently.

Auto Associative Kernel Regression (AAKR) is a regression method, based on the multivariate inferential kernel regression (Wand & Jones, 1995). AAKR is often applied for the development of normal behaviour models.

A.2.2 Classification

The task of predicting a qualitative (i.e. categorical) value based on a number of inputs is called classification. An example of classification is presented in Figure A.6. Given a number of input values and their corresponding class, lines that separate different classes are calculated on the feature space.

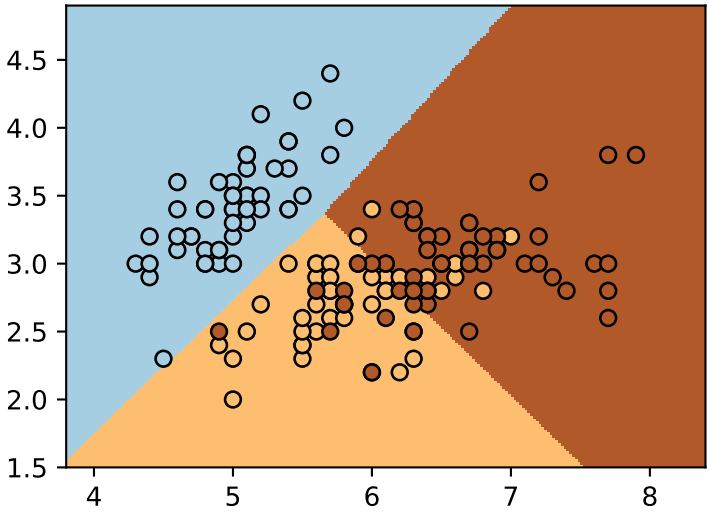

Figure A.6: illustrative example of linear, multi-class classification. Given a number of input values and their corresponding class, lines that separate different classes are calculated on the feature space.

Linear Discriminant Analysis (LDA), as discussed in the unsupervised learning section, is a method that identifies a linear combination of features that can be used to distinguish a dataset. This feature permits the algorithm to be used for linear classification purposes (Rao, 1973).

A.2.3 Regression & Classification

Algorithms that can be used in both regression and classification tasks are presented below.

Support Vector Machines (SVMs) in their simplest form constitute a two-class classifier in cases where the two classes are linearly separable. SVMs work by deriv-

ing the optimal hyperplane, i.e. the hyperplane that offers the widest possible margin between instances of the two classes. Their functionality can be extended by the introduction of a non-linear kernel, allowing them to learn non-linear mappings, i.e., classify between non-linearly separable classes (Theodoridis & Koutroumbas, 2009b). SVMs can also be built as regressors (Smola & Schölkopf, 2004). Support Vector Reggressors (SVRs) work in a similar way, this time trying to fit a hyperplane that accurately predicts the target values of training samples within a margin of tolerance.

Least Squares Support Vector Machine (LLSVM) represents the least-squares version of SVM where obtaining a model entails solving a set of linear equations, instead of quadratic programming as is the case for classical SVMs (Suykens & Vandewalle, 1999).

The Artificial Neural Network (ANN) and the Deep Neural Network (DNN) are computing systems, inspired by the way biological nervous systems work. Various ANN architectures exist, offering superior performance at many machine learning tasks, including classification and regression. ANNs are extremely versatile as they can accurately model complex non-linear behaviours. ANNs are based on an interconnected group of connected units (neurons) where each connection between these units transmits a signal from one to another, when the linear combination of its inputs exceeds some threshold (Russell & Norvig, 2010). The receiving unit can process that signal and then pass it on to the next unit.

Consequently, depending on the number of layers implemented, ANNs can be classified as shallow and deep. Whilst no formal rule exists to separate shallow and deep neural networks (Schmidhuber, 2015), usually networks that have more than 1 hidden layer are considered deep. As the number of layers increases, the model can "learn" more non-linear behaviours. At the same time, training becomes more computationally expensive and the risk of overfitting the dataset also increases.

Extreme Learning Machine (ELM) is a type of ANN where the parameters of the hidden nodes do not require tuning, first proposed by Huang et al. (2006). Due to this, training ELM is often thousands of times faster than training ANNs using traditional tuning, i.e. backpropagation.

Kernel-based Extreme Learning Machine (K-ELM) is an extension of ELM, where a kernel is used to add a non-linearity feature similarly to kernel-based SVM models (Huang, 2014).

K-Nearest Neighbours (KNN) is a supervised learning method applied to both classification and regression methods. In classification applications, the input observation is assigned the most common class observed in the k nearest neighbouring observations. Accordingly, in regression applications, the average of the k nearest neighbouring observations is used as the model output.

Decision Tree (DT) models partition the feature space into rectangles and learn a simple (e.g. constant) model in each of those (Hastie et al., 2009). DTs do not produce a continuous output in the traditional sense. Instead, these models are trained on a training set whose outputs lie on a continuous range. Their output ends up being the mean value of the training set observations that reside in the same node.

One of the most common methods for tree-based regression is Classification And Regression Trees (CART) (Breiman et al., 1984). In this case, the original feature space is split into two regions, selecting the split point and dependent variable (feature) to obtain the best model fit (Hastie et al., 2009). This is performed recursively, until the activation of a stopping rule.

Random Forests (RFs) are based on the bagging (i.e. bootstrap aggregating) meta-algorithm, where a number of de-correlated decision trees are trained based on the available training set (Ho, 1995, 1998). Then, the output of the random forest model is calculated by averaging the results of individual decision trees. Bagging refers to the

development of homogeneous weak learners that learn independently and in parallel and then combination of their individual output as model output.

Boosting trees is an equivalent to RFs, where the boosting meta-algorithm is used (Drucker & Cortes, 1996). In this case, homogeneous weak learners are built sequentially, with each model depending on the performance of previous models, and the combination of their individual output is returned as model output.

Hierarchical Support Vector Machine (H-SVM) is the hierarchical version of SVM where a decision tree is trained with individual SVM models as end nodes (Chen et al., 2004). Applications usually focus on multi-class classification problems but H-SVM models are also occasionally used for regression (Bellocchio et al., 2013).

A.2.4 1-class Classification

A special case of classification is one-class classification where all data points available during model training are assumed to belong to one class and new observations are classified as either belonging to that class or not. An illustrative example of one-class classification is presented in Figure A.7.

One Class Support Vector Machine (OC-SVM) follow the same working principles as normal SVM models derived for classification purposes with the caveat that, in this case, the model aims to derive the hyperplane that separates all training points from the origin instead of a hyperplane that separates points belonging to different classes (Schölkopf et al., 1999). In cases where the hyperplane is spherical instead of planar, the method is called Support Vector Data Description (SVDD) (Tax & Duin, 2004).

One-Class Extreme Learning Machine (OC-ELM) trains a model that accepts observations that belong to the original class and rejects all others, based on ELM principles. This is achieved through a suitable distance function (Leng et al., 2015).

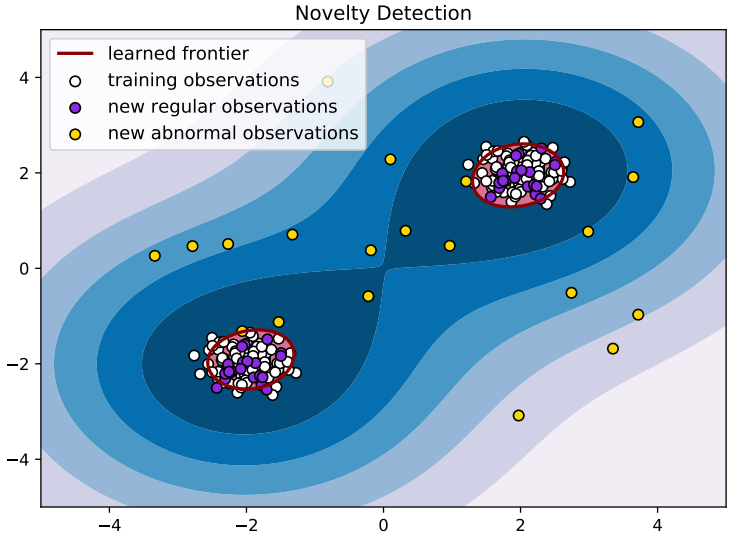

Figure A.7: illustrative example of one-class classification based on a non-linear one-class SVM model.

Autoencoders (AEs) constitute a special type of ANN, often used for dimensionality reduction purposes (Goodfellow et al., 2016; Kramer, 1991). This functionality can be extended to anomaly detection through the evaluation of the quality of the developed reconstruction (Sakurada & Yairi, 2014).

Bibliography

Acomi, N. & Acomi, O. C. (2014). Improving the Voyage Energy Efficiency by Using EEOI. *Procedia - Social and Behavioral Sciences*, *138*, 531–536.

Allianz. (2019). *Safety and Shipping Review 2019* (tech. rep.). Allianz Global Corporate & Specialty. Munich, Germany.

Amozegar, M. & Khorasani, K. (2016). An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines. *Neural Networks*, *76*, 106–121.

Armstrong, V. N. & Banks, C. (2015). Integrated approach to vessel energy efficiency. *Ocean Engineering*, *110*, 39–48.

Azur, M. J., Stuart, E. A., Frangakis, C. & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*.

Bellman, R. (2003). *Dynamic programming*. Dover Publications.

Bellocchio, F., Borghese, N. A., Ferrari, S. & Piuri, V. (2013). Hierarchical support vector regression. *3d surface reconstruction: Multi-scale hierarchical approaches* (pp. 111–142). Springer New York.

Bergstra, J. S., Bardenet, R., Bengio, Y. & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization.

Beşikçi, E. B., Kececi, T., Arslan, O. & Turan, O. (2016). An application of fuzzy-AHP to ship operational energy efficiency measures. *Ocean Engineering*, *121*, 392–402.

Bialystocki, N. & Konovessis, D. (2016). On the estimation of ship's fuel consumption and speed curve: A statistical approach. *Journal of Ocean Engineering and Science*, *1*(2), 157–166.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.

Brandsæter, A., Vanem, E. & Glad, I. K. (2019). Efficient on-line anomaly detection for ship systems in operation. *Expert Systems with Applications*, *121*, 418–437.

Breiman, L. (1998). Arcing classifiers. *Annals of Statistics*, *26*(3), 801–849.

Breiman, L., Friedman, J. H., Olshe, R. A. & Stone, C. J. (1984). *Classification and Regression Trees*. Taylor & Francis.

BS EN 13306. (2017). Maintenance — Maintenance terminology.

Burges, C. J. C. (2010). Geometric methods for feature extraction and dimensional reduction – a guided tour. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 53–82). Springer.

Casteleiro-Roca, J. L., Quintián, H., Calvo-Rolle, J. L., Corchado, E., del Carmen Meizoso-López, M. & Piñón-Pazos, A. (2016). An intelligent fault detection system for a heat pump installation based on a geothermal heat exchanger. *Journal of Applied Logic*, *17*, 36–47.

Chandroth, G. (2004). Condition monitoring: the case for integrating data from independent sources. *Proceedings of IMarEST-Part A-Journal of Marine Engineering and Technology*, *2004*(4), 9–16.

Chang, C. C. & Lin, C. J. (2011). LIBSVM: A Library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*.

Chang, C.-C. & Lin, C.-J. (2001). Training ν-support vector classifiers: Theory and algorithms. *Neural Comput.*, *13*(9), 2119–2147.

Chawla, N. V., Japkowicz, N. & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, *6*(1), 1–6.

Chen, Y., Crawford, M. & Ghosh, J. (2004). Integrating support vector machines in a hierarchical output space decomposition framework. *IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004*, *2*, 949–952.

Cichowicz, J., Theotokatos, G. & Vassalos, D. (2015). Dynamic energy modelling for ship life-cycle performance assessment. *Ocean Engineering*, *110*, 49–61.

Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J. & Anguita, D. (2018a). Condition-Based Maintenance of Naval Propulsion Systems with supervised Data Analysis. *Ocean Engineering*, *149*, 268–278.

Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J. & Anguita, D. (2018b). Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback. *Reliability Engineering & System Safety*, *177*, 12–23.

Clarke, B., Fokoue, E. & Zhang, H. H. (2009). *Principles and Theory for Data Mining and Machine Learning*. Springer New York.

Clarkson PLC. (2018). *Annual Report 2018: Smarter decisions. Powered by intelligence.* (tech. rep.). London, Great Britain.

Coraddu, A., Oneto, L., Baldi, F. & Anguita, D. (2017). Vessels fuel consumption forecast and trim optimisation: A data analytics perspective. *Ocean Engineering*, *130*, 351–370.

Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D. & Figari, M. (2016). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, *230*(1), 136–153.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Croarkin, C., Tobias, P., Filliben, J. J., Hembree, B., Guthrie, W., Trutna, L. & Prins, J. (Eds.). (2018). *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH.

Dai, H., Cao, J., Wang, T., Deng, M. & Yang, Z. (2019). Multilayer one-class extreme learning machine. *Neural Networks*, *115*, 11–22.

Deloitte. (2015). *Shipping smarter - IoT opportunities in transport and logistics* (tech. rep.). Deloitte University Press. London, Great Britain.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Dikis, K., Lazakis, I., Michala, A. L., Raptodimos, Y. & Theotokatos, G. (2017). Dynamic risk and reliability assessment for ship machinery decision making. *Risk, Reliability and Safety: Innovating Theory and Practice - Proceedings of the 26th European Safety and Reliability Conference, ESREL 2016*, 110.

Dikis, K. & Lazakis, I. (2016). Dynamic risk and reliability assessment of ship machinery and equipment. *Proceedings of the International Offshore and Polar Engineering Conference*, *2016-Janua*, 969–976.

Dinwoodie, I. (2014). *Modelling the operation and maintenance of offshore wind farms* (Doctoral dissertation). University of Strathclyde, Department of Electronic and Electrical Engineering.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*.

Drucker, H. & Cortes, C. (1996). Boosting Decision Trees. In D. S. Touretzky, M. C. Mozer & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8* (pp. 479–485).

Emovon, I., Norman, R. A. & Murphy, A. J. (2018). Hybrid MCDM based methodology for selecting the optimum maintenance strategy for ship machinery systems. *Journal of Intelligent Manufacturing*, *29*(3), 519–531.

Equasis. (2006). *The World Merchant Fleet in 2005 - Statistics from Equasis* (tech. rep.). European Maritime Safety Agency (EMSA). Saint Malo, France.

Equasis. (2009). *The World Merchant Fleet in 2008 - Statistics from Equasis* (tech. rep.). European Maritime Safety Agency (EMSA). Saint Malo, France.

Equasis. (2012). *The World Merchant Fleet in 2011 - Statistics from Equasis* (tech. rep.). European Maritime Safety Agency (EMSA). Saint Malo, France.

Equasis. (2015). *The World Merchant Fleet in 2014 - Statistics from Equasis* (tech. rep.). European Maritime Safety Agency. Saint Malo, France.

Equasis. (2018). *The World Merchant Fleet in 2017 - Statistics from Equasis* (tech. rep.). European Maritime Safety Agency (EMSA). Saint Malo, France.

Faulstich, S., Hahn, B. & Tavner, P. J. (2011). Wind turbine downtime and its importance for offshore deployment. *Wind Energy*, *14*(3), 327–337.

Fedele, L. (2011). *Methodologies and Techniques for Advanced Maintenance*. Springer London.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Fitch, E. C. (1992). *Proactive Maintenance for Mechanical Systems*. Elsevier Science.

Geman, S., Bienenstock, E. & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, *4*(1), 1–58.

Girdhar, P. & Scheffer, C. (2004). *Practical machinery vibration analysis and predictive maintenance*. Newnes.

Glantz, S. & Slinker, B. (2000). *Primer of applied regression & analysis of variance*. McGraw-Hill Education.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT Press.

Guillén, A. J., González-Prida, V., Gómez, J. F. & Crespo, A. (2016). Standards as Reference to Build a PHM-Based Solution. In K. T. Koskinen, H. Kortelainen, J. Aaltonen, T. Uusitalo, K. Komonen, J. Mathew & J. Laitinen (Eds.), *Proceedings of the 10th world congress on engineering asset management (wceam 2015)* (pp. 207–214). Springer International Publishing.

Gurney, K. (1997). *An introduction to neural networks*. UCL Press.

Hanachi, H., Liu, J., Kim, I. Y. & Mechefske, C. K. (2019). Hybrid sequential fault estimation for multi-mode diagnosis of gas turbine engines. *Mechanical Systems and Signal Processing*, *115*, 255–268.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Hotelling, H. (1936). Relations Between Two Sets of Variates. *Biometrika*, *28*(3/4), 321.

Hountalas, D. (2000). Prediction of marine diesel engine performance under fault conditions. *Applied Thermal Engineering*, *20*(18), 1753–1783.

Hsu, C.-W., Chang, C.-C. & Lin, C.-J. (2010). A practical guide to support vector classification.

Hu, L.-Y., Huang, M.-W., Ke, S.-W. & Tsai, C.-F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, *5*(1), 1304.

Huang, G.-B. (2014). An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognitive Computation*, *6*(3), 376–390.

Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1-3), 489–501.

IMO. (2016). Resolution MEPC.278(70) - Amendments to the annex of the protocol of 1997 to amend the international convention for the prevention of pollution from ships, 1973, as modified by the protocol of 1978 relating thereto amendments to MARPOL Annex VI.

Inmarsat. (2018). *Industrial IoT on land and at sea - Inmarsat Research Programme 2018* (tech. rep.). Inmarsat plc. London, Great Britain.

International Monetary Fund. (2019). *World Economic and Financial Surveys: World Economic Outlook Database. April 2019.* (tech. rep.). Washington, D.C., United States.

Jang, J. (1991). Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm. *Proceedings of the 9th National Conference on Artificial Intelligence*.

Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, *23*(3), 665–685.

Jardine, A. K., Lin, D. & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.

Jung, D. (2019). Engine Fault Diagnosis Combining Model-based Residuals and Data-driven Classifiers. *IFAC-PapersOnLine*, *52*(5), 285–290.

Kaski, S. & Kohonen, T. (1996). Exploratory Data Analysis By The Self-Organizing Map: Structures Of Welfare And Poverty In The World (1996). *Neural Networks*

in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets, 498–507.

Khan, S. S. & Madden, M. G. (2010). A survey of recent trends in one class classification. In L. Coyle & J. Freyne (Eds.), *Artificial intelligence and cognitive science* (pp. 188–197). Springer Berlin Heidelberg.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*.

Kriegel, H.-P., Kröger, P. & Zimek, A. (2010). Outlier detection techniques. *Tutorial at KDD*, *10*.

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Kusiak, A. & Li, W. (2011). The prediction and diagnosis of wind turbine faults. *Renewable Energy*, *36*(1), 16–23.

Kwan, C., Xu, R. & Zhang, X. (2003). Fault detection and identification in aircraft hydraulic pumps using MCA. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *36*(5), 1137–1142.

Lamaris, V. & Hountalas, D. (2010). A general purpose diagnostic technique for marine diesel engines – Application on the main propulsion and auxiliary diesel units of a marine vessel. *Energy Conversion and Management*, *51*(4), 740–753.

Laouti, N., Sheibat-Othman, N. & Othman, S. (2011). *Support vector machines for fault detection in wind turbines* (Vol. 44). IFAC.

Law, A. M. (2006). How to build valid and credible simulation models. *Proceedings of the 38th Conference on Winter Simulation*, 58–66.

Lazakis, I. & Ölçer, A. (2016). Selection of the best maintenance approach in the maritime industry under fuzzy multiple attributive group decision-making environment. *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, *230*(2), 297–309.

Lazakis, I., Turan, O. & Aksu, S. (2010). Increasing ship operational reliability through the implementation of a holistic maintenance management strategy. *Ships and Offshore Structures*, *5*(4), 337–357.

Leahy, K., Hu, R. L., Konstantakopoulos, I. C., Spanos, C. J., Agogino, A. M. & O'Sullivan, D. T. (2018). Diagnosing and predicting wind turbine faults from scada data using support vector machines. *International Journal of Prognostics and Health Management*, *9*(1), 1–11.

Leng, Q., Qi, H., Miao, J., Zhu, W. & Su, G. (2015). One-Class Classification with Extreme Learning Machine. *Mathematical Problems in Engineering*, *2015*.

Li, P., Liu, L. & Gong, H. (2010). The Research of the Intelligent Fault Diagnosis Optimized by ACA for Marine Diesel Engine. Springer, Berlin, Heidelberg.

Lin, Y.-H., Fang, M.-C. & Yeung, R. W. (2013). The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic elements. *Applied Ocean Research*, *43*, 184–194.

Lu, F., Jiang, J., Huang, J. & Qiu, X. (2017). Dual reduced kernel extreme learning machine for aero-engine fault diagnosis. *Aerospace Science and Technology*, *71*, 742–750.

Lu, R., Boulougouris, E., Banks, C. & Incecik, A. (2015). A semi-empirical ship operational performance prediction model for voyage optimization towards energy efficient shipping. *Ocean Engineering*, *110*, 18–28.

Lundh, M., Garcia-Gabin, W., Tervo, K. & Lindkvist, R. (2016). Estimation and Optimization of Vessel Fuel Consumption. *IFAC-PapersOnLine*, *49*(23), 394–399.

Luo, F.-L., Unbehauen, R. & Cichocki, A. (1997). A Minor Component Analysis Algorithm. *Neural Networks*, *10*(2), 291–297.

MAN B&W Diesel A/S. (2004). *Instruction Book 'Operation' for 50-108MC/MC-C Engines* (2nd ed.).

Mao, W., Rychlik, I., Wallin, J. & Storhaug, G. (2016). Statistical models for the speed prediction of a container ship. *Ocean Engineering*, *126*, 152–162.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, *405*(2), 442–451.

Mehta, B. R. & Reddy, Y. J. (2015). *Industrial process automation systems: design and implementation*. Butterworth-Heinemann.

Meng, Q., Du, Y. & Wang, Y. (2016). Shipping log data based container ship fuel efficiency modeling. *Transportation Research Part B: Methodological*, *83*, 207–229.

MEPC. (2019). *2019 Guidelines on consistent implementation of 0.50% sulphur limit under MARPOL Annex VI* (tech. rep.). IMO Marine Environment Protection Committee (MEPC).

Mohanty, A. R. (2017). *Machinery condition monitoring: principles and practices* (1st). CRC Press.

Moore Stephens LLP. (2018). *Future operating costs report, October* (tech. rep.). Moor Stephens International Limited. London, Great Britain.

Moreno-Gutiérrez, J., Calderay, F., Saborido, N., Boile, M., Rodríguez Valero, R. & Durán-Grados, V. (2015). Methodologies for estimating shipping emissions and energy consumption: A comparative analysis of current methods. *Energy*, *86*, 603–616.

Neale, M. & Associates. (1979). *A Guide to the Condition Monitoring of Machinery*. H.M. Stationery Office.

Nowlan, F. S. & Heap, H. F. (1978). *Reliability-Centered Maintenance* (AD/A066 57, tech. rep.). United Airlines. San Fransisco, California.

Orozco, R., Sheng, S. & Phillips, C. (2018). Diagnostic Models for Wind Turbine Gearbox Components Using SCADA Time Series Data. *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*, 1–9.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Perera, L. P. (2016). Marine Engine Centered Localized Models for Sensor Fault Detection under Ship Performance Monitoring. *IFAC-PapersOnLine*, *49*(28), 91–96.

Perera, L. P. & Mo, B. (2018). Ship performance and navigation data compression and communication under autoencoder system architecture. *Journal of Ocean Engineering and Science*, *3*(2), 133–143.

Petersen, J. P., Winther, O. & Jacobsen, D. J. (2012). A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions. *Ship Technology Research*, *59*(1), 64–72.

Prpić-Oršić, J., Vettor, R., Faltinsen, O. M. & Guedes Soares, C. (2016). The influence of route choice and operating conditions on fuel consumption and CO2 emission of ships. *Journal of Marine Science and Technology*, *21*(3), 434–457.

Randall, R. B. & Antoni, J. (2011). Rolling element bearing diagnostics-A tutorial. *Mechanical Systems and Signal Processing*, *25*(2), 485–520.

Rao, C. R. (1973). *Linear statistical inference and its applications* (2nd). Wiley.

Raptodimos, Y. & Lazakis, I. (2018). Using artificial neural network-self-organising map for data clustering of marine engine condition monitoring applications. *Ships and Offshore Structures*, *13*(6), 649–656.

Roh, M.-I. (2013). Determination of an economical shipping route considering the effects of sea state for lower fuel consumption. *International Journal of Naval Architecture and Ocean Engineering*, *5*(2), 246–262.

Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Pearson.

Sakurada, M. & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. *ACM International Conference Proceeding Series*, *02-December-2014*, 4–11.

Santos, P., Villa, L. F., Reñones, A., Bustillo, A. & Maudes, J. (2015). An SVM-based solution for fault detection in wind turbines. *Sensors (Switzerland)*, *15*(3), 5627–5648.

Schlechtingen, M., Santos, I. F. & Achiche, S. (2013). Using data-mining approaches for wind turbine power curve monitoring: A comparative study. *IEEE Transactions on Sustainable Energy*, *4*(3), 671–679.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J. & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, *13*(7), 1443–1471.

Schölkopf, B., Smola, A. J., Williamson, R. C. & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*(5), 1207–1245.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J. & Platt, J. (1999). Support vector method for novelty detection. *Proceedings of the 12th International Conference on Neural Information Processing Systems*, 582–588.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O. & Hemingway, H. (2014). Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using MICE: A CALIBER Study. *American Journal of Epidemiology*, *179*(6), 764–774.

Simonsen, M., Walnum, H. & Gössling, S. (2018). Model for estimation of fuel consumption of cruise ships. *Energies*, *11*(5), 1059.

Sina Tayarani-Bathaie, S. & Khorasani, K. (2015). Fault detection and isolation of gas turbine engines using a bank of neural networks. *Journal of Process Control*, *36*, 22–41.

Smola, A. J. & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Soualhi, A., Medjaher, K. & Zerhouni, N. (2015). Bearing health monitoring based on hilbert-huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, *64*(1), 52–62.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J. & Nenadic, G. (2019). Machine learning methods for wind turbine condition monitoring: A review. *Renewable Energy*, *133*, 620–635.

Stopford, M. (2009). *Maritime economics*. Routledge.

Suykens, J. A. & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Tax, D. M. & Duin, R. P. (2004). Support Vector Data Description. *Machine Learning*, *54*(1), 45–66.

Theissler, A. (2017). Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. *Knowledge-Based Systems*, *123*, 163–173.

Theodoridis, S. & Koutroumbas, K. (2009a). *Pattern recognition* (4th). Academic Press.

Theodoridis, S. & Koutroumbas, K. (2009b). *Pattern recognition* (4th). Academic Press.

Touret, T., Changenet, C., Ville, F., Lalmi, M. & Becquerelle, S. (2018). On the use of temperature for online condition monitoring of geared systems – A review. *Mechanical Systems and Signal Processing*, *101*, 197–210.

Trodden, D. G., Murphy, A. J., Pazouki, K. & Sargeant, J. (2015). Fuel usage data analysis for efficient shipping operations. *Ocean Engineering*, *110*, 75–84.

Tsitsilonis, K. M. & Theotokatos, G. (2018). A novel systematic methodology for ship propulsion engines energy management. *Journal of Cleaner Production*, *204*, 212–236.

UNCTAD. (2018). *Review of Maritime Transport 2018* (tech. rep.). United Nations Conference on Trade and Development. United Nations.

Van Hulse, J., Khoshgoftaar, T. M. & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. *Proceedings of the 24th International Conference on Machine Learning*, 935–942.

Veneti, A., Makrygiorgos, A., Konstantopoulos, C., Pantziou, G. & Vetsikas, I. A. (2017). Minimizing the fuel consumption and the risk in maritime transportation: A bi-objective weather routing approach. *Computers & Operations Research*, *88*, 220–236.

Vettor, R. & Guedes Soares, C. (2016). Development of a ship weather routing system. *Ocean Engineering*, *123*, 1–14.

Walther, L., Rizvanolli, A., Wendebourg, M. & Jahn, C. (2016). Modeling and Optimization Algorithms in Ship Weather Routing. *International Journal of e-Navigation and Maritime Economy*, *4*, 31–45.

Wand, M. & Jones, M. (1995). Summary for Policymakers. In Intergovernmental Panel on Climate Change (Ed.), *Climate change 2013 - the physical science basis* (pp. 1–30). Cambridge University Press.

Wang, L., Zhang, Z., Long, H., Xu, J. & Liu, R. (2017). Wind Turbine Gearbox Failure Identification with Deep Neural Networks. *IEEE Transactions on Industrial Informatics*, *13*(3), 1360–1368.

Wang, S., Ji, B., Zhao, J., Liu, W. & Xu, T. (2018). Predicting ship fuel consumption based on LASSO regression. *Transportation Research Part D: Transport and Environment*, *65*, 817–824.

Watzenig, D., Sommer, M. S. & Steiner, G. (2009). Engine state monitoring and fault diagnosis of large marine diesel engines. *Elektrotechnik und Informationstechnik*, *126*(5), 173–179.

Widodo, A. & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, *21*(6), 2560–2574.

Wong, P. K., Zhong, J., Yang, Z. & Vong, C. M. (2016). Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis. *Neurocomputing*, *174*, 331–343.

World Bank Group. (2019). *Commodity Markets Outlook, April* (tech. rep.). World Bank. Washington, DC.

Wu, X. & Srihari, R. K. (2003). New ν-support vector machines and their sequential minimal optimization. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 824–831.

Xi, P. P., Zhao, Y. P., Wang, P. X., Li, Z. Q., Pan, Y. T. & Song, F. Q. (2019). Least squares support vector machine for class imbalance learning and their applic-

ations to fault detection of aircraft engine. *Aerospace Science and Technology*, *84*, 56–74.

Xu, L., Oja, E. & Suen, C. Y. (1992). Modified Hebbian learning for curve and surface fitting. *Neural Networks*, *5*(3), 441–457.

Xu, Q. H. & Shi, J. (2006). Fault diagnosis for aero-engine applying a new multi-class support vector algorithm. *Chinese Journal of Aeronautics*, *19*(3), 175–182.

Yao, Z., Ng, S. H. & Lee, L. H. (2012). A study on bunker fuel management for the shipping liner services. *Computers & Operations Research*, *39*(5), 1160–1172.

You, C. X., Huang, J. Q. & Lu, F. (2016). Recursive reduced kernel based extreme learning machine for aero-engine fault pattern recognition. *Neurocomputing*, *214*, 1038–1045.

Zhang, H., Chen, H., Guo, Y., Wang, J., Li, G. & Shen, L. (2019). Sensor fault detection and diagnosis for a water source heat pump air-conditioning system based on PCA and preprocessed by combined clustering. *Applied Thermal Engineering*, *160*(October 2017), 114098.

Zhao, Y. P., Huang, G., Hu, Q. K., Tan, J. F., Wang, J. J. & Yang, Z. (2019). Soft extreme learning machine for fault detection of aircraft engine. *Aerospace Science and Technology*, *91*, 70–81.

Zhao, Y. P., Song, F. Q., Pan, Y. T. & Li, B. (2017). Retargeting extreme learning machines for classification and their applications to fault diagnosis of aircraft engine. *Aerospace Science and Technology*, *71*, 603–618.

Zhao, Y.-P., Wang, J.-J., Li, X.-Y., Peng, G.-J. & Yang, Z. (2019). Extended least squares support vector machine with applications to fault diagnosis of aircraft engine. *ISA Transactions*, (40), 27–29.

Zis, T. P., Psaraftis, H. N. & Ding, L. (2020). Ship weather routing: A taxonomy and survey. *Ocean Engineering*, *213*, 107697.