International Journal of Information Management Data Insights 3 (2023) 100178

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

Enabling digital twins in the maritime sector through the lens of AI and industry 4.0

Dimitrios Kaklis[a,*], Iraklis Varlamis[c], George Giannakopoulos[b], Takis J. Varelas[d], Constantine D. Spyropoulos[b]

[a] Department of Informatics and Telematics, Harokopio University of Athens, NCSR Demokritos, Danaos Shipping Co., Omirou 9, Tavros, Athens, 17778, Greece
[b] Institute of Informatics & Telecommunications, NCSR Demokritos, Patr. Gregoriou E and 27 Neapoleos Str, Agia Paraskevi, Athens, GR-15341, Greece
[c] Department of Informatics and Telematics, Harokopio University of Athens, Patr. Gregoriou E and 27 Neapoleos Str, Agia Paraskevi, Athens, Omirou 9 Tavros, GR 17778, Greece
[d] Danaos Shipping Co., Akti Kondyli 14, Piraeus, GR-18450, Greece

ARTICLE INFO

Keywords:
Fuel oil consumption estimation
Digital twin
Splines
Quadratic estimators
Delaunay triangulation
Time-series forecasting
Unsupervised clustering
Ensemble learning
Deep learning
Least squares optimization

ABSTRACT

Sustainability and environmental compliance in ship operations is a prominent research topic as the waterborne sector is obliged to adopt "green" mitigation strategies towards a low emissions operational blueprint. Fuel-Oil-Consumption (FOC) estimation, constitutes one of the key components in maritime transport information systems for efficiency and environmental compliance. This paper deals with FOC estimation in a more novel way than methods proposed in literature, by utilizing a reduced-sized feature set, which allows predicting vessel's Main-Engine rotational speed (RPM). Furthermore, this work aims to place the deployment of such models in the broader context of a cutting-edge information system, to improve efficiency and regulatory adherence. Specifically, we integrate B-Splines in the context of two Deep Learning architectures and compare their performance against state-of-the-art regression techniques. Finally, we estimate FOC by combining velocity measurements and the predicted RPM with vessel-specific characteristics and illustrate the performance of our estimators against actual FOC data.

1. Introduction

Port logistics can often be a complex and challenging process, with various barriers and obstacles that can impede the smooth flow of goods and cargo (Sarkar & Shankar, 2021). One of the key challenges in port logistics is route optimization, which involves finding the most efficient and cost-effective way to transport goods from the port to their destination. Optimal ocean route planning is strongly connected to the fuel oil consumption (FOC) of sea vessels and the minimization the CO_2 emissions that reduce cost and the environmental footprint of Shipping. Among other factors, this approach aids jointly towards the efficient and robust ship tracking, weather forecasting, and emission control. The existing spatiotemporal data-driven solutions are employed upon a multitude of features, from vessel tracking devices and structural properties of the ship, to features that capture weather and internal machinery condition. Vessel monitoring and tracking can be performed using Synthetic Aperture Radar (SAR) images and data from the Automatic Identification System (AIS) (Zhao, Ji, Xing, Zou, & Zhou, 2014) or other surveillance systems (Chen et al., 2020). The spatial dimension that completes the vessel monitoring focuses on local conditions, such as the waves and currents that affect the cost of overseas movement, while the temporal aspect monitors environmental and ship-system conditions and examines how they evolve with time.

The multitude of data-driven methods that are employed in modern Information Systems for vessel monitoring and route optimization fuse features from multiple sensors onboard (Filippopoulos et al., 2022). However, there are many cases where ships share the minimum monitoring information (e.g. AIS messages, noon reports), essentially offering little more than their position and (implicitly) their speed over ground (SOG). FOC is highly affected by the velocity of the vessel and the weather conditions of the voyage. Furthermore it is closely related to the rotational speed, of the vessel's Main Engine (M/E) (Avgouleas, 2008), measured in Revolutions Per Minute (RPM). The latter determines the rotational speed of the propeller that produces the required thrust. This implies that the optimal route problem can be significantly optimized and restructured, if a good predictive model for RPM is made available, and the vessel's velocity can be a useful feature in this direction. Throughout the manuscript, scalar values of speed V are utilized in sections concerning model employment and experimental evaluation. The term velocity is used only as a reference to the reader, to indicate that

* Corresponding author.
E-mail address: dk.drc@danaos.gr (D. Kaklis).

https://doi.org/10.1016/j.jjimei.2023.100178
Received 31 December 2022; Received in revised form 10 April 2023; Accepted 11 April 2023

originally, the acquired measurements of speed, are vectors, entailing the scalar value and the direction of the vessel.

This work focuses on the relation between spatiotemporal derived velocity measurements (V) and *quasi steady* operational variables like RPM to form a reference $V \rightarrow RPM$ model, without *initially* including weather or other operational features in the training phase. The over-ground velocity of the vessel V can be easily estimated by harvesting its <latitude, longitude> coordinates that are collected every few seconds from the vessel's Automatic Identification System (AIS) (Pallotta, Vespe, & Bryan, 2013). With a model that accurately predicts RPM we can then calculate an initial estimate of FOC, using SFOC (Specific Fuel Oil Consumption; gr/kWh) that is provided by the M/E manufacturer, the vessel's speed, and some vessel particulars, following a computational approach. The $V \rightarrow RPM$ model for a particular vessel type can be used as a reference basis for more vessels of the same type, and transfer learning techniques in conjunction with vessel-specific variables to fine-tune and apply the same model to more vessels.

We predict RPM using only the vessel speed information, by generating different, velocity-based, vector representations in different implicit environmental and operational settings of a vessel. Actually, we posit the argument, that partitioning the input space and creating appropriate *local* models connecting velocity-based patterns to RPM estimates can improve the prediction performance over an *overall* learned estimation model. To support our claim we develop and evaluate a number of different approaches. Motivated by the work in (Rippa, 1990), which connects **splines** with **Delaunay Triangulations** (**DTri**), we first employ an analytical method for the estimation of RPM, based on the construction of piecewise polynomial representations over a DTri that partitions an appropriately defined 2D vector space of velocity measurements. We innovatively apply a spatiotemporal derived method (Wang, Wang, Lai, & Gao, 2020) to a domain that exclusively consists of temporal correlations (in the form of splines) between the velocity observations and RPM. We thus demonstrate how splines can be combined with the partitioning of the input space in regions of similar vessel behavior, and how deep learning techniques can be used to improve the prediction of RPM.

Instantaneous changes in RPM caused either by maneuvering or due to severe weather fluctuations, result in a nonlinear relationship between V and RPM, as varying RPM values correspond to different ranges of speed. Hence, we consider that the representation of a given moment in time needs to take into account previous values, so that it can better express the context of a specific instance in time. For this purpose, we examine how different representations of the input space (from sequential raw data to moving average values) affect the estimation performance. We also examine whether partitioning the input space and creating local sub-models improve the estimation of a global model for the whole input space.

The aforementioned proposed methodology is complemented with the employment of a big data tool initially demonstrated in (Kaklis, Giannakopoulos, Varlamis, Spyropoulos, & Varelas, 2019), which continuously harvests operational data (STW[1], RPM, etc.) from the vessel in real-time. This cutting-edge integrated framework incorporates the latest technological advances to capture, process, and analyze vessels' data in order to improve efficiency, sustainability, and rule compliance. Particularly, we demonstrate the conceptualization and materialization of a big data application suite that exploits the IoT (Internet of Things) and AI (Artificial Intelligence) advancements and technologies, to construct a "digital replica" of the en-route vessel. The presented framework aims to facilitate the employment of a holistic representation of the vessel's operational state offering a multi-modal approach comprising real-time data aggregation-processing modules, as well as a physics-informed model repository to monitor validate and project the FOC of the fleet and therefore the corresponding emissions.

[1] Speed Through Water

The proposed platform is thoroughly described in § 3.

To this end, this work contributes to the research field of sustainable maritime operations by:

- Demonstrating a multi-modal ICT[2] framework adapted to the needs of the maritime sector, which offers an efficient, safer, and self-sustainable operational blueprint;
- Utilizing core modules of the aforementioned digital ecosystem to propose and evaluate different input space partitioning methods (from clustering to triangulation);
- Examining how these methods affect prediction performance when combined with different estimators;
- Performing experiments on real data, acquired from IoT installments on board container-ship vessels, to evaluate the performance and behavior of the different proposed system variations;
- Demonstrating how the estimated RPM can offer sufficiently good FOC estimation when combined with vessel-specific reference (SFOC) curves;
- Finally, we propose promising ways to empower established deep sequential models with splines to improve prediction performance over sequential data.

As the maritime sector gradually but steadily transcends to Industry 4.0 era, the materialization and integration of Management Information Systems (MIS) constitutes one of the fundamental pillars for all parties attached to the waterborne sector (ports, suppliers, charterers, shipowners), to increase competitiveness and enhance operational efficiency. The holistic digitization of the vessel, through a variety of state of the art MIS, will pave the way for a transparent and greener sea transportation by continuously assessing and improving the operational, financial and environmental plausibility of all the possible emerging technologies related to the vessel's life cycle (energy saving devices, alternative fuels, new build designs, retrofitting solutions, Remotely Operated Vehicles (ROVs) & Autonomous Vehicles (AVs) for predictive maintenance, etc.). The realization of a digitized version of the vessel, leveraging state of the art technologies (Big Data Analytics Grover & Kar, 2017, Process Mining Kouzari, Sotiriadis, & Stamelos, 2023, Deep Learning Kar & Kushwaha, 2021; Marc, 2022) with integrated systems tools and connectivity (Deepu & Ravi, 2021; Jain, Seeja, & Jindal, 2021; Venkatachalam & Ray, 2022) will enable the short and long term competitiveness of maritime stakeholders, by securing efficiency in terms of operational costs (OPEX, CAPEX, TCE[3]) and environmental compliance (CII, EEDI[4] indicators).

The paper is organized as follows: Section 2 provides a general introduction to the pertinent literature on smart transportation systems, a thorough examination of routing optimization techniques adopted in the maritime sector, as well as a short introduction regarding use of splines in NNs. Section 3 outlines the architecture and the main design principles of a prototype Digital Twin framework for the maritime sector, supporting the proposed methodologies, in the context of this work, in terms of data aggregation, curation, model training, validation, and deployment. Section 4 explains how partially linear models can be used for time-series prediction over a partitioned domain and proposes a two-step process for predicting RPM based on velocity measurements. Section 5 demonstrates an extension of the proposed methodology by introducing two deep neural network architectures, one feed-forward and one recurrent, which incorporate theory derived from B-Spline approximation, implicitly or explicitly, to employ a Black-Box and a Grey-Box approach respectively. Section 6 describes the dataset utilized for the evaluation process and interprets the experimental results. Finally, Section 8 provides the main conclusions of this work and outlines next steps in the context of future work.

[2] Information Communication Technology

[3] Operational Expenditure, Capital Expenditure, Time Charter Equivalent

[4] Carbon Intensity Indicator, Energy Efficiency Design Index

2. Related work

Pertinent literature regarding smart and efficient transportation in the industry (supply chain optimization, autonomous vehicle, logistics management, etc.) as well as regarding societal frameworks (traffic management, public transportation networks, suburban mobility), concerns a variety of multi-constraint optimization methods varying from Genetic Algorithms (Wei, Yang, & Huang, 2021), Simulated annealing (Fermani, Rossit, & Toncovich, 2021) and Particle Swarm Optimization (Chondrodima, Georgiou, Pelekis, & Theodoridis, 2020), to AI integrated decision making using Reinforcement Learning (Bai, Shangguan, Cai, & Chai, 2019) or NLP based approaches (Garg, Kiwelekar, Netak, & Ghodake, 2021). The aforementioned practices and methodologies attempt to exploit the proliferation of IoT devices and state-of-the-art communication networks (5G) to build self-contained Information Hubs and provide a sustainable, safer and cost-effective transportation.

In the maritime sector, the optimization criteria adopted in the context of the ship routing problem deal with the minimization of voyage time, fuel consumption (or Fuel Oil Consumption, FOC) and voyage risk. The approaches, which have appeared so far in the literature, can be classified into three broader categories: i) *Vessel-based optimization*, which aims in optimizing a given route with respect to vessel characteristics, e.g., vessel speed, main-engine rotational speed, draft, trim and sea-keeping behavior: roll, heave and pitch motions, Coraddu, Oneto, Baldi, & Anguita (2017), Roh & Lee (2018); ii) *Environmental-based optimization*, which aims in optimizing a given route by taking into account environmental conditions, e.g., wind (speed, direction), wave (height, frequency, direction), currents. Krata & Szlapczynska (2018), Kim & Kim (2017); iii) *Holistic optimization* that combine the two previous approaches in a common context (Golias, Saharidis, Boile, Theofanis, & Ierapetritou, 2009; Varelas et al., 2013; Vettor & Soares, 2015; Walsh & Bows, 2012).

Most of FOC theoretical calculations, found in pertinent naval engineering literature, are based on the Admiralty coefficient which is extensively used by marine practitioners and engineers in the estimation of the power that is required for a new build design to attain the required speed, and is given by the formula:

$$k = \left(\sqrt[3]{\Delta^2} \cdot V^3\right) / P_S \tag{1}$$

where is the displacement (tn) of the vessel, V is the desired speed and P_s is the so-called EHP (Effective Horse Power) (kW).

The techniques employed for FOC prediction, which are based on vessel characteristics and environmental conditions, can be further grouped into three sub-categories (Coraddu et al., 2017): i) ***White Box or Analytical approaches*** where analytical equations and approximation methods (e.g. Computational Fluid Dynamic equation - CFD), are exploiting a variety of vessel specific variables and hydrodynamic principles to model the added resistance of the hull of a specific vessel. The admiralty constant as well as the resistance constant $RL/\Delta V^2$, where R is the total resistance and L is the length overall of the vessel, proposed by Telfer (1963) has been utilized in the past to replicate the hydrodynamic behavior of new ship designs by using only the design point values of the corresponding parameters (speed (V), displacement (Δ), length (L), breadth (B), draught (T)), rather than the whole spectrum of the operational domain. Thus it can be easily inferred that the purpose of the Admiralty constant was to provide an initial baseline to compare the hydrodynamic performance of different ships in their respective design conditions, rather than monitor their operational state. This makes evident that the Admiralty constant was neither intended nor demonstrated to be a suitable operational hydrodynamic performance indicator for a ship. ii) ***Data-driven approaches*** that combine vessel-trajectory data, gathered from sensors, satellites (AIS data) or Noon Reports, with Machine and Deep-Learning algorithms. These techniques are ranging from simple Regression analysis, using stand-alone models like Support Vector Regression (SVR), Lasso Regression (LR), Polynomial Regression, to ensemble non-parametric schemes like Random Forest regression (RF), Decision Trees or AdaBoost where the approximation power of each model is combined appropriately in order to infer the underlying function. Gkerekos, Lazakis, & Theotokatos (2019), Wang, Ji, Zhao, Liu, & Xu (2018) and Deep Learning approaches. Ahlgren & Thern (2018), Miyeon, Noh, Shin, O-Kaung Lim, & Cho (2018), Kaklis et al. (2022a). Some methods dealt also with the problem of deteriorating performance as new batches of data corresponding to arbitrary distributions are introduced to the estimator (Kaklis, Varlamis, Giannakopoulos, Spyropoulos, & Varelas, 2022b); iii) ***Hybrid approaches*** that combine Machine Learning methods (also known as black-box models - *BBM*), with analytical methods (known as white-box models - *WBM*), such as the equations of motion of a freely floating body moving with constant forward speed, in order to increase the prediction accuracy. The resulting models are known as grey-box models (*GBM*) (Coraddu et al., 2017; Kaklis et al., 2019).

Some studies experimented also with baseline sequential neural networks, by applying a dropout in the weights in order to achieve better generalization error (Gkerekos & Lazakis, 2020), or by tuning the number of hyper parameters (i.e., learning rate, number of neurons, number of layers, activation function, etc.) and utilizing brute force methods like grid search (Jeon et al., 2018; Papandreou & Ziakopoulos, 2020; Savitha, Al Mamun et al., 2017). Zhu, Zuo, & Li (2020) employed a Recurrent NN (Neural Network) in order to estimate FOC, but without further research as far as the NN architecture.

The use of splines in Neural Networks (NN) is not new, nevertheless, it has not yet been used in the FOC prediction problem. Campolucci, Capperelli, Guarnieri, Piazza, & Uncini (1996) introduced NN with Generalized Sigmoidal neurons (GS-neurons) that employ the Catmull-Rom spline adaptive activation function. Their systems were capable to simulate the behavior of Nonlinear Dynamic Systems. Recently, a combined B-spline-NN and ARX (autoregressive-exogenous) Model for representing the nonlinear and linear static parts, respectively, of a dynamic actuation system was presented in Folgheraiter (2016). Instead of using splines as custom adaptive piecewise activation functions, Lin (2012) used cubic splines to calculate the weights in the neurons of a Deep Learning model that identifies the road surface power spectrum density. Similarly, Chen, Hong, Khalaf, Morfeq, & Alotaibi (2015) proposed a nonlinear equalization approach, which is based on a B-spline NN to simulate the non-linearity of a High Power Amplifier.

However, all the previous work evaluated the performance of spline-powered NN on synthetic (Campolucci et al., 1996; Folgheraiter, 2016) or simulated (Chen et al., 2015; Lin, 2012) datasets and not on a real-world dynamic system. In this work, we apply similar principles in the task of RPM prediction from Velocity time-series and evaluate the performance of our methods on a real dataset acquired from multiple container ship vessels (Living Labs) that were operated on a timeframe of one year in order to elicit the appropriate set of requirements and KPIs for the realization and assessment of the specific use case (FOC approximation). In addition to this, the previous approaches result in much simpler networks than the ones we examine in this work and they are restricted to the predictive capabilities of the spline models. Even the existing multi-input adaptive BSNNs (Folgheraiter, 2016) still lack the ability to optimize the contribution of each spline to the final prediction for each input.

The majority of the approaches found in literature regarding Routing Optimization and FOC estimation in the maritime sector concern the implementation of standalone services in the sense that they are employing isolated information silos that lack the support mechanisms and enhancement of a centralized Information Hub that exploits the upsurge of IoT and Industry 4.0 advancements, to train, validate and update these services in real-time.

Furthermore they are usually tested on a single vessel and therefore lack the generalization capabilities of models evaluated in a variety of ships that are able to adjust and adapt to the underlying function that describes the relationship between FOC and each specific vessel, continuously, by exploiting the vast amount of data collected by IoT in-

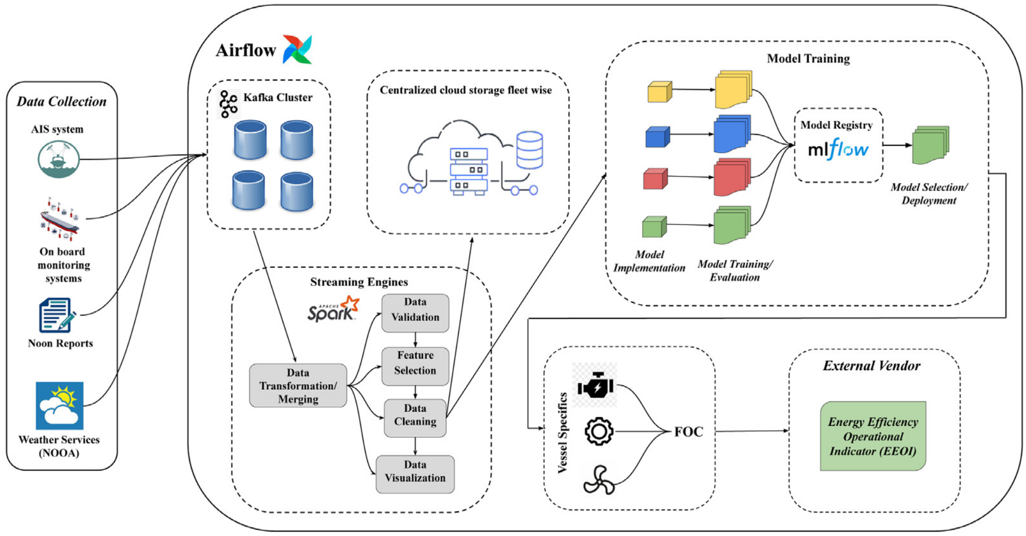


Fig. 1. The ”concentrated” workflow from data collection to FOC estimation.

stallments. Frameworks and technological advancements regarding the continuous monitoring of the vessel are inextricably linked with the emerging concept of the so-called Digital Twin in the shipping industry, as they employ a digital replica of the en-route vessel that is able to simulate-project and validate in real time the majority of the operational procedures.

3. Prototype of a digital - twin framework

Management Information Systems (MIS) is a discipline that involves the use of information technology to support and improve the management and decision-making processes of an organization. Through the use of various tools and techniques, MIS helps organizations collect, store, and analyze data in order to gain insights and make informed decisions. Closely related to MIS is the emerging concept of the so-called *Digital Twin*, first introduced in Grieves, 2014, that represents a virtual representation of a physical system or process that can be used to simulate and analyze its behavior. A Digital Twin, adapted to the needs of the maritime sector, constitutes a virtual holistic representation of the vessel that spans its life-cycle and is updated from near to real-time data, utilizing simulation, machine learning and reasoning to help in decision-making, sensing and control actuation. By combining core structural properties of traditional MIS and digital twins, organizations can gain a better insight of their internal operations and pave the way for a fully automated and fault tolerant decision making procedure, improving substantially their efficiency and effectiveness.

In Kaklis et al. (2022a) we demonstrated a Big Data Analytics system adapted to the needs of the maritime sector. The proposed framework incorporates a variety of state-of-the-art streaming tools for real-time analysis of vessel data as well as tools for continuous integration/deployment (CI/CD) of ML/DL models regarding operational optimization, causal analysis, and event recognition. By utilizing the company's existing in-house infrastructure concerning Edge-Headquarter (EDGE-HQ) communication between the vessel and the office, we can incorporate the aforementioned pipeline in a broader data acquisition network in order to aggregate, synchronize and process data coming from the vessel in real-time. The resulting platform (see Fig. 1) constitutes a prototype version of a virtual replica of the en-route vessel (Digital Twin (DT) framework) that aims to assist shipowners to achieve efficiency in fleet management with tangible benefits in terms of emission reduction, environmental compliance and protection of crew safety onboard.

In the following, we describe how we can minimize human involvement in such systems by introducing fully automated administrative workflows following the design principles that govern Expert Systems, in order to create a centralized digital framework to manage efficiently office - vessel communication, model simulation as well as reasoning and decision making.

In the context of this work, the aforementioned framework was adapted to the needs of the proposed methodology, by acquiring storing and analyzing the speed of the vessel as well as the corresponding RPM of the ship's M/E. Furthermore leveraging the parallel processing capabilities of streaming tools with a set of orchestration frameworks we were able to build an adaptive model repository. As an entity this is part of a broader component of the DT framework referenced as Knowledge Hub (KH) (see Section 3.1).

Mainly, the proposed DT framework consists of the following components, as shown in Fig. 2:

- IoT backbone suite & Third party Connectors: Data acquisition layer
- Knowledge Hub: Processing - Orchestration - Computing - Deployment layer
- Main GUI: Visualization layer - Dashboard
- Decision Support System (DSS): KPIs identification, evaluation of possible solutions
- Edge Computing: Sensing & Control actuation layer, Requirements & Refinements elicitation

3.1. Knowledge hub

The core module of the proposed DT framework is the Knowledge Hub (KH). The hub incorporates a variety of multi-disciplinary approaches regarding data provision, re-usability and curation as well as state-of-the-art frameworks for model versioning and deployment. It constitutes a holistic approach that aims to create an adaptive and versatile observatory for the shipping industry that comprises structured methodologies for interconnecting each use case with the appropriate data, processing algorithms, and simulation models. All these are joined together adequately, facilitating towards the decarbonization of the maritime sector. More specifically, the core modules of KH as presented in Fig. 2 are **CMDS** (Centralized Messaging Distribution System), the **Monitoring platform** as well as the **Model Repository**. These adaptive multi-purpose systems combine a variety of edge technologies, in a consolidated approach that aims to streamline, standardize, as well as offer a layer of transparency and encapsulation of the various processes and scenarios involved in the DT ecosystem. The appropriate orchestration of the three layers related to: 1) the harmonization of information 2) storage of different versions of models as well as 3) the decision-making

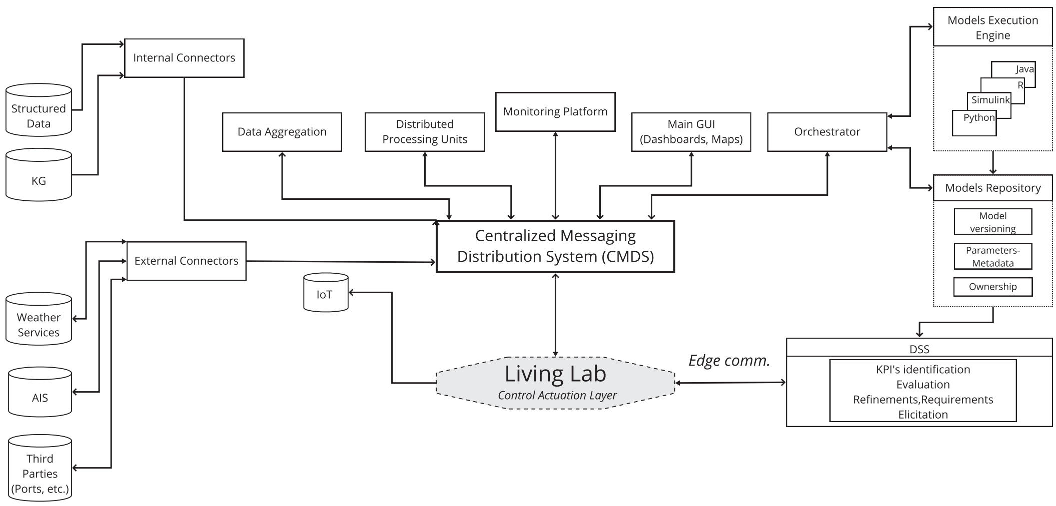


Fig. 2. Holistic representation of the envisaged DT framework.

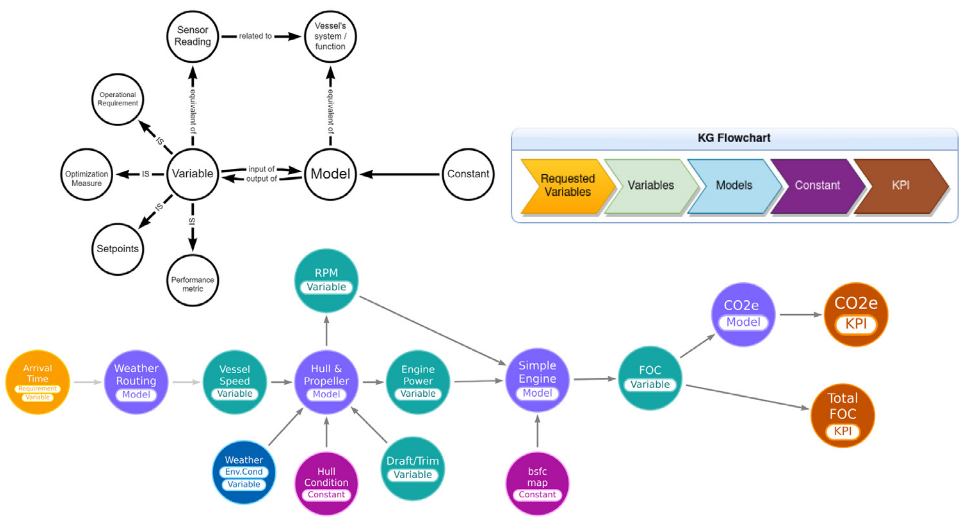


Fig. 3. The streamline procedure adapted for the FOC estimation use case.

center, attribute to the DT ecosystem the characteristic of gradual cognitive capability facilitating towards the development of self-sustained administrative workflows that aim to continuously refine, update and optimize the system as a whole.

Fig. 3 illustrates a multi-modal streamlined procedure, stored in KH, adapted to the task of FOC estimation.

The main goal of KH is to largely simplify and standardize the way the various tools and services provided by the DT's ecosystem are operating and communicating with each other, following the standards of an ICT (Information Communication Technology) framework. The general streamlined procedure is based on:

- Data Curation from bias and noise,
- Data Processing and feature selection, and
- Model Versioning & Deployment.

Data processing focuses on determining the most important features, depending on the use case, and data curation accounts for removing the bias (outliers, faulty measurements) from the bulk of data collected in real time from IoT installments. The resulting feature set is utilized accordingly in the training process of data-driven models or to analytically calculate theoretical models inferred from pertinent naval engineering literature.

As showcased in Fig. 3, the general procedure can be adapted to the needs of a specific use case by associating relevant operational variables with the appropriate algorithms and simulation models and applying them to practice. The following sections focus on the specific case of FOC estimation, through RPM prediction from vessel speed, and provide a sensitivity analysis framework that aims to evaluate each approach individually in order to find an ideal RPM-FOC predictive scheme.

4. Efficient predictions over time-series data utilizing clustering

The relation between RPM and V can be described via a partially linear function with nonlinear segments over time as showed in Kaklis et al. (2019) and also derived from operational data as demonstrated in Fig. 4. This non-linearity between V and RPM in different time segments on a vessel's trip may be attributed to two reasons: i) ambient weather conditions (such as wind speed or/and wave heights), ii) sudden changes in the propeller shaft (RPM) that don't translate immediately to a corresponding change in velocity. This complex dependence of RPM on V is evident in Fig. 4, which depicts data from measurements on a real vessel over a time period of approximately 10,000 minutes. The figure indicates that that the relationship between the two variables involves linear as well as nonlinear segments.

Based on the above remark, it is legitimate to assume that a mix of linear and nonlinear models can be used for the prediction of RPM from a series of V measurements in each segment of the vessel trip.

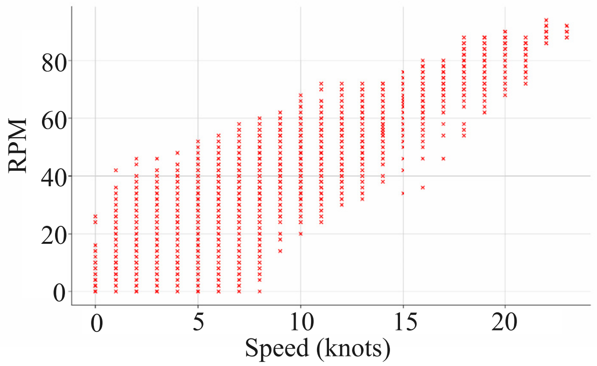


Fig. 4. Propeller operating conditions in different speed ranges.

Furthermore, utilizing naval-architecture literature on calculating the total resistance (R_T) of a vessel, we can extract preliminary insights regarding the relationship between V and R_T and consequently get an initial conclusion on the relationship between V and RPM from standard principles. More specifically it is known from basic ship theory that the total resistance R_T is calculated as follows:

$$R_T = \frac{1}{2} C_T \, \rho \, S \, V^2, \tag{2}$$

where C_T is the non-dimensional total resistance coefficient ($0 < C_T < 1$), ρ is the sea water density, V is the vessel speed and S is the wetted surface of the hull. The power P_s absorbed by the ship propulsion system, shaft horsepower (SHP) in the marine engineering terminology, is expressed as:

$$P_s = \frac{R_T \, V}{\eta_p}, \tag{3a}$$

where η_p is the propulsion coefficient, usually ranging from 55% to 75%, defined as:

$$\eta_p = \eta_{gear} \, \eta_{shaft} \, \eta_{propeller} \, \eta_{hull}. \tag{3b}$$

A high level interpretation of our work is depicted in Fig. 1. In the first step of the pipeline we determine the source of the data, generated either from the AIS or from on board monitoring installments. Depending on the source we proceed with the Data Transformation - Validation and Cleaning steps where we extract the necessary features (V, RPM in this work) and remove the noise (outliers, errors) with a streamlined procedure described more thoroughly in Kaklis et al. (2019). After the processing step is complete, we advance with the clustering phase where we partition our input space in different operational states, in terms of velocity and the corresponding RPM, utilizing a clustering technique (e.g. K-Means, DTC). In the training step that follows, we assemble and train a collection of predictors (either local estimators that take into account the temporal correlations discovered in the previous step or Deep Neural Network architectures) in order to optimize the prediction of M/E's RPM. Finally, after evaluating the models we select the best-fit in terms of accuracy and generalization error and we proceed by displaying an initial approximation of the FOC utilizing vessel's specific variables.

The current work focuses on the methodology for building and using the predictive RPM-FOC models, and does not put emphasis on the implementation aspects that outline the DT framework. The model building phase comprises an initial step of data partitioning in segments of similar $V \rightarrow RPM$ relation and a second step of ML/DL model training. Similarly, the inference phase comprises a step where the input data are mapped to the appropriate partition and a second step where the respective model is used to get a prediction.

In order to improve the predictive ability of our model we include the impact of the memory window for each instantaneous velocity $V(t_i)$ that is formed from the k previous values of $V(t_i)$. So a formal definition of the problem of predicting RPM values based on the continuously monitored ship velocity over ground V can be defined as follows:

Definition 1. Given the vessel's speed for $k+1$ consecutive points in time $\{t_0, \dots, t_k\}$, (i.e., a time series $(V_0, \dots, V_k)$ of velocity values), find a function $f(V_0, \dots, V_k) : R^k \rightarrow R$, which estimates the engine's RPM at the subsequent moment t_{k+1}.

4.1. RPM prediction in two steps

The principles that govern the $V - RPM$ relationship as described above, can be utilized as the basis of a generic and modular prediction process that facilitate different tasks and implementations. More specifically we introduce a two-step process that includes: i) *partitioning* of the input space, and ii) *finding an approximation* of the mapping function $RPM(V, \bar{V}_N)$ in the sense of Definition 1, through an aggregate of quadratic (or higher order) polynomials, each of which models one of the partitions generated in the first step. Here $\bar{V}_N$ indicates the average of V over a memory window that involves N previous velocity measurements. We elaborate on the above idea to support different partitioning approaches and locally-aware approximation functions in the following paragraphs. Henceforth, we interchangeably use the terms "clustering" and "partition" in order to denote the split of the input space into subspaces.

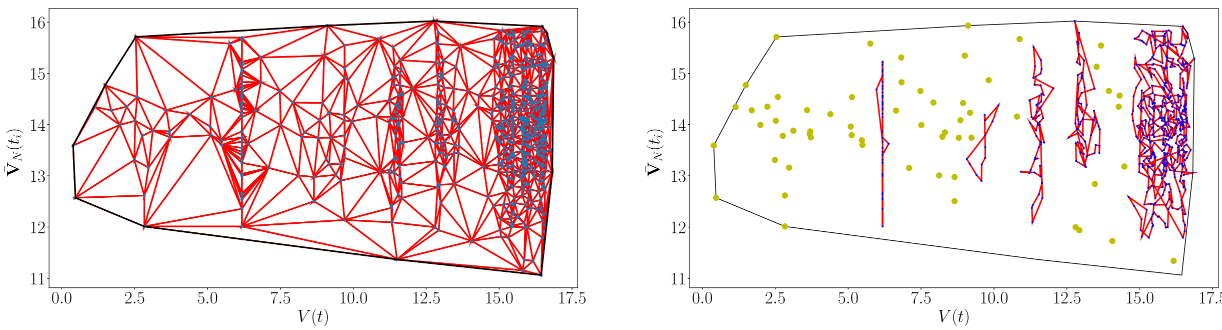
In order to partition our input space $(V, \bar{V}_N)$ in areas of distinguishable distributions (different vessel operational states) taking into account temporal autocorrelation and heterogeneity, we employ the methods described below:

- **K-means clustering (KM)** (Lloyd, 1957; MacQueen, 1967) is a vector quantization method, originating from the field of signal processing, that is widely used for data clustering. Its main aim is to partition the observations (vectors) into K clusters so that each observation belongs to the cluster of its nearest centroid (i.e. representative vector of the cluster).
- **Triangulation clustering (DC)** (Eldershaw & Hegland, 1997) first partitions the training space in triangles using a triangulation-based method. The main reason for opting in favor of DTri among other triangulation techniques, is that is closely connected with the so-called *Delaunay Configurations* that, as stated in Neamtu (2007), is linked with a multivariate extension of the univariate B-splines used in this work for approximation. All these properties combined, secure the construction of piecewise linear surfaces of minimal Dirichlet energy as stated again in Neamtu (2007) resulting in concentrated homogeneous clusters of data with low variance.

By selecting a cut-off value p (used to determine the neighboring points from the adjacency list of each candidate vector $[V, \bar{V}_N]$), we can find for each point in the training space its neighboring vertices in the resulting graph. By applying a Depth-First-Search (DFS) algorithm it is possible to find isolated subgraph components recursively as depicted in Fig. 5(b), which shows the resulting clusters for the pointset in Fig. 5(a).

The basic idea behind clustering with triangulation is that it defines the cluster in a much broader manner, than, e.g., K-means, being able to cluster observations in non-spherical neighborhoods. Also K-means, in its general definition used here, does not seem able to detect outliers. In contrast to the K-means algorithm, DT-based clustering, as depicted in Fig. 5(b), is able to detect and remove outliers from clusters resulting in more "reliable" clusters.

In order to utilize appropriately the partitioning of the input space we continue by employing and testing a set of regression methods that take into consideration the heterogeneous distributions of the clusters generated in the first step. These regression methods correspond to either baseline ML models or deep learning architectures each one modeling the partitioned data in a different way. We infer a Spline estimator for each one of the partitions of the input space, to employ a more adaptive prediction scheme than a single generic approximation function, over the whole space. Regression models that employ multivariate



(a) Convex hull and Delaunay Triangulation (DT) of a planar training pointset

(b) Clustering outcome after applying DFS on the DTri of Fig. 5(a); points belonging to the same cluster are connected with red linear segments; green points indicate outliers.

Fig. 5. Delaunay Triangulation and resulting clustering.

adaptive regression splines (Friedman, 1991) capture non-linearities using piecewise polynomial functions.

In the context of this work two additional regression techniques were used as a baseline alongside Spline Regression: Linear Regression (Montgomery, Peck, & Vining, 2021) and Random Forest Regression (Cootes, Ionita, Lindner, & Sauer, 2012).

4.2. *Averaging global and local effects:* White Box modeling

In order to appropriately utilize theory concerning total resistance calculation (Eq. (2)) as well as input space partitioning, we demonstrate in this section a physics inspired procedure to construct different B-Spline interpolants on different appropriately constructed $2D$ clusters. These clusters are modeled on the basis of the two alternatives, namely, K-Means or DTC, and they correspond to the reduced feature set $[V(t_i), \bar{V}_N(t_i)]$ of the vessel's speed as formulated in previous paragraphs. Diving into pertinent maritime engineering literature in order to extract further useful information regarding the relationship between the power absorbed by the propulsion system and the speed V of the vessel, we identified another important indicator outlining the hydrodynamic performance of the vessel, namely the so-called modern *Admiralty coefficient* C_{ADM}, adopted by the *International Towing Tank Conference* (ITTC, 2021), defined as below:

$$C_{ADM} = \frac{\Delta^{2/3} V^3}{P_s}, \tag{4}$$

where Δ is the displacement of the vessel in tons. Values of the dimensional coefficient C_{ADM} usually range from 400 to 600, the higher the value the more economical the vessel. From (4) and (3a) we readily get the following admiralty-coefficient-based estimators for the resistance R_T and power P_s:

$$R_T = \eta_p\, C_{ADM}^{-1}\, \Delta^{2/3} V^2, \tag{5a}$$

and

$$P_s = C_{ADM}^{-1}\, \Delta^{2/3} V^3. \tag{5b}$$

For small variations of V around a design velocity one can assume that C_{ADM} in (5) remains constant. This is not anymore the case when a ship should operate efficiently for more than one design speeds or along an interval of speeds. In the spirit of Gupta, Taskar, Steen, & Rasheed (2021) one could propose the following generalization of (5b):

$$P_s = c\Delta^q V^r, \tag{6}$$

where the factor c and the exponents q and r are treated as unknowns. Noting that:

$$\ln(P_s) = \ln(c) + n\ln(\Delta) + m\ln(V), \tag{7}$$

in Gupta et al. (2021) it is proposed to work with linear polynomials on the logarithmic scale and use the available measurement data for estimating the unknowns c, q and r.

Instead, our approach is to train a cubic multivariate B-Spline model for the chosen QoI (Quantity of Interest), e.g., P_s, RPM, FOC, using as variables the adopted feature set. Choosing our B-spline model to be cubic stems from the fact that estimators (3a) and (5) depend on V^r, $r = 2, 3$. As a result our model is able to exactly reconstruct any QoI that depends linearly ($r = 1$), quadratically ($r = 2$) or cubically ($r = 3$) with respect to a feature variable x. If the dependence is different from $x^r, r = 1, 2, 3$, the cubic B-Spline model remains efficient due its approximation capacity characterized by quadratic rate of convergence towards any smooth (C^2) QoI(x) as the window h of the B-spline knot sequence decreases. In Fig. 6 below we depict the resulting curve from fitting with cubic B-Splines in a snapshot of the $2D$ space of $[V] \to RPM$ and the corresponding surface from training the multivariate equivalent B-Spline interpolant in the 3D space of $[V(t_i), \bar{V}_N(t_i)] \to RPM$ observations.

By training and eventually aggregating, multiple, physics informed, B-Spline interpolants on different clusters we are able to construct a multi modal adaptive scheme that aims to balance ideally the trade-off between goodness of fit and smoothness by exploiting core properties of third degree (Cubic) Spline interpolation (e.g continuous 2nd derivatives).

5. Combining splines with deep learning

In order to jointly take advantage of the Delaunay Triangulation properties and the capabilities of nonlinear models and their combinations to predict relations that are not linear in all segments, we propose a black-box alternative to the WB approach of § 4. More specifically, we use Delaunay Triangulation to partition the input space in conjunction with NN (Neural Networks) to approximate the mapping function. The different input space partitions and the respective subsets of training instances are used to train individual predictors. Since NN allow us to learn how to combine multiple predictors using additional layers, we examine several NN-architectures that consolidate the approximation power of multiple predictors. We evaluate the performance of such methods, by moving gradually from a method that trains separately different estimators and *learns* how to ideally combine these estimators in a weighted average ensemble, to the recurrent equivalent of such methods, namely the LSTM (Long-Short Term -Memory) network.

In the reviewed literature (Campolucci et al., 1996; Fey, Eric Lenssen, Weichert, & Müller, 2018; Folgheraiter, 2016; Zhengyu, Donald S., Williams, & Xiangning, 2007) the connection between B-Spline functions and NN is implemented mostly through custom activation functions that utilize the piecewise estimators in the final layer of the networks in order to filter out the information passed in the next training cycle to update the weights of the model. In this work, the basic

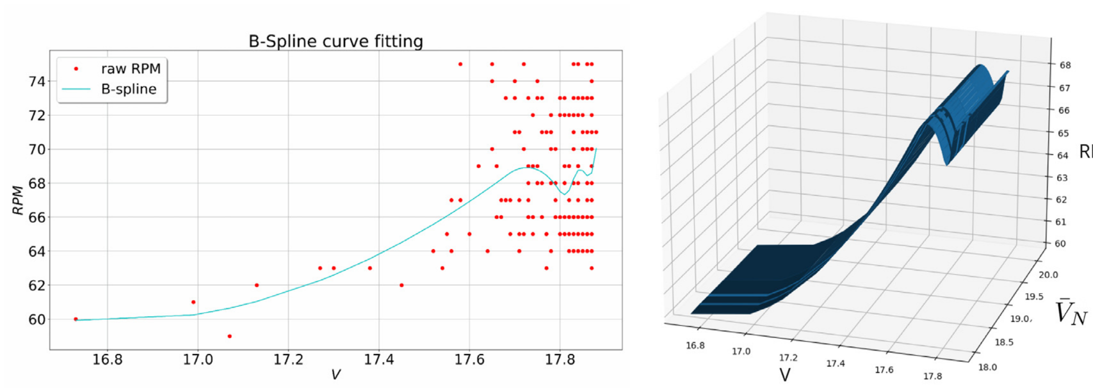


Fig. 6. $2D$ Bi-variate & $3D$ Multivariate B-Spline fitting.

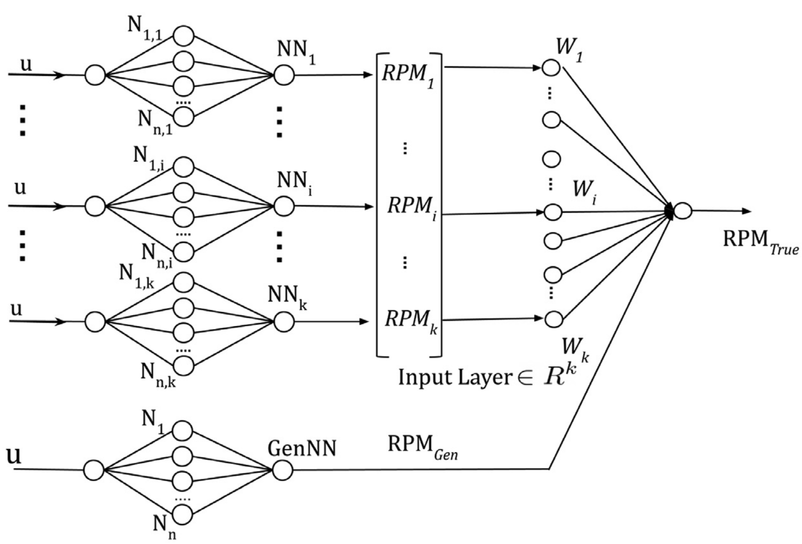


Fig. 7. The NN-architecture that predicts the weights to combine all models (local and general) in a weighted average sum - *wAvgNN*.

NN consists of one input and three hidden layers, each one consisting a number of neurons specified by extracting information from a pre-trained regression model. A rectified linear unit (ReLU) is used as the activation function of each layer. ReLU is defined as $y(x) = \max(0, x)$, and is a function that – in contrast to other activation functions – back-propagates the larger percentage of the error on the output to update the neuron weights. A stochastic gradient descent process, namely the AdaGrad-optimizer of the Keras framework, has been used to optimize the weights for the neural network.

The various implemented NN-architectures, which are described in detail in the following subsections, are the following:

- A NN-architecture, termed *wAvgNN*, that combines different estimators trained on different clusters, in a weighted average, in which the weights are learned after training.
- A second NN-architecture, termed *BaseLSTM*, that employs a basic Long-Short Term Memory Neural Network, with multiple parallel time-series as input.
- Finally, an extension of the aforementioned recurrent NN that attempts to utilize spline interpolation by introducing a Spline-informed recurrent NN, termed *SplineLSTM*.

5.1. Consolidating models in an ensemble NN: Black box modeling

The approaches presented so far are based on the initial partitioning of the training data. They classify any new sample to the most appropriate cluster, by using a similarity function, and consequently use the respective local effect model - NN_i in Fig. 7 - for predicting RPM.

In this section, we propose a NN-architecture that utilizes a layer in order to find a weighted average of all the local models and the global model. This weighted average quantifies the percentage of participation of the local models and the global model to the final prediction of RPM in a data driven manner. The extra layer - as depicted in Fig. 7- aims to facilitate the cognitive curve of the neural proposed, eventually paving the way for a more robust, regularized prediction scheme.

The proposed predictive scheme is depicted in Fig. 7 and consists of many local models trained on different parts of the dataset as described in previous sections (for V and $\bar{V}_N$) in the input layer, which is densely connected with the k neurons of the output layer.

In this methodological approach a global spline-regression model (*OverallNeural*) is pre-trained on the complete dataset, and is used to define the number of clusters, which equals the number of knots of the Spline. Then, using the predictions $(RPM_i, \dots, RPM_k)$ of the local models we train a NN to find the weight vector $[W_i, \dots, W_k]$ used to calculate the (weighted) average of the local models.

Each training instance $[V(t_i), \bar{V}_N(t_i)]$ is evaluated at every local effect model (NN_i) and results in estimating one RPM_i value. The combined outputs form a vector ($\in R^k$) of RPM, where each RPM_i value corresponds to the ith domain of the local model NN_i function that is trained on a particular cluster. The NN architecture can, thus, be used to learn the weight that must be assigned to the prediction of the ith local effect NN in the weighted average. The target weights of each local effect NN are calculated and updated in the back-propagation process as follows:

$$W_i = \sqrt{|RPM_i - RPM_{true}|^2} \tag{8}$$

where RPM_{true} is the actual value for this training instance.

At the final stage we estimate RPM by calculating average between two terms:

- the weighted average of each RPM_i value evaluated at each cluster with the weights predicted from the NN,
- the RPM_{Gen} prediction the $GenModel$ trained in the dataset as a whole.

We expect that the above process provides a more data-aware version of the averaging operator, making the architecture more adaptive to different settings.

5.2. Combining LSTM neural networks with splines

The aforementioned BBM technique (§ 5.1) assumes that the input space is two-dimensional, combining velocity measurements with average velocity values estimated over a time window before each instantaneous measurement. Since velocity is actually a time series, we decide to evaluate the performance of deep (recurrent) NN-architectures, and more specifically LSTMs on this time-series view of the data. Long short-term memory (LSTM) is an artificial Recurrent Neural Network (RNN) architecture (Hochreiter & Schmidhuber, 1997), that has been utilized for time-series prediction (Hua et al., 2019), classification (Karim, Majumdar, Darabi, & Chen, 2017) and anomaly detection (Malhotra, Vig, Shroff, & Agarwal, 2015). Unlike standard feed-forward neural networks, LSTM also contains feedback connections and can process single data points (such as images), as well as entire sequences of data (such as speech, video or trajectories). The relative insensitivity of LSTM's with respect to the length of gaps between important events in a time series is an advantage compared to RNN's, hidden Markov models, and other sequence learning methods in numerous applications. To this end, we utilize an LSTM NN-architecture for the prediction of RPM values from consecutive velocity measurements in a time window, as described in the following paragraphs.

Using the same window length N as in the non-recurrent architectures described previously – and instead of using $(V, \bar{V_N})$ – we utilize the original sequence of N measured velocities from time $t_0 - n, 0 \leq n \leq N$, where t_0 is the time of interest and N is a specific time-window length (time-lag) that consists of prior information for the state of the vessel (here velocity) that is useful to estimate the RPM value at time t_0. The resulting input vector at a time t_i is of dimension $N+1$ and has the form: $[V_i, V_{i1} \dots V_{iN}]$. Then, given a sequence of moments through time, we get the following correspondence between the input and the output (RPM) vectors:

$$\begin{bmatrix} [V_1, V_{11} \dots V_{1N}] \\ \vdots \\ [V_i, V_{i1} \dots V_{iN}] \\ \vdots \\ [V_M, V_{M1} \dots V_{MN}] \end{bmatrix} \longrightarrow \begin{bmatrix} RPM_1 \\ \vdots \\ RPM_i \\ \vdots \\ RPM_M \end{bmatrix} \tag{9}$$

where N is the step to form the time window vectors for the hidden LSTM units as we described before and M is the sample size. The output value to predict is RPM. This baseline LSTM approach is further referenced as *BaseLSTM*.

5.2.1. SplineLSTM*: Grey box modeling*

Grey Box models, a term first introduced in Bohlin (1992), are a combination of theoretical models (WBM) and data driven approaches (BBM). WBM's are usually simpler models in terms of computational complexity and they attempt to calculate the target variable of the approximation problem at hand from a theoretical standpoint by applying the contextually appropriate law of physics that governs each problem. In their simplest implementation GBM's are attempting to integrate prior knowledge extracted from a theoretical model into a BBM. They do this by incorporating two approaches: a naive approach (N-GBM), where the output of the WBM is utilized as a new feature in the BBM training process, and a more advanced approach (A-GBM), where a regularization term is introduced in the loss function of the BBM. Another approach would be to induce in a core module (layer, activation function, etc.) information extracted from the WBM in order to facilitate the cognitive curve (learning rate) of the of the network. Our aim in this section is to combine a black box deep learning architecture with the theoretical backbone of naval engineering and develop a more efficient GBM that could better approximate resistance on the wetted area of the vessel and therefore RPM.

In alignment with the above comments, N-GBM approaches refer mostly to methods that incorporate in the training process of a BBM, the output of a theoretical baseline model (WBM). Usually, an N-GBM allows the creation of a new dataset with an enriched input vector:

$$\mathcal{D}_n = \left\{ \left(\begin{bmatrix} x_1 \\ f_{WBM(x_1)} \end{bmatrix}, y_1 \right), \dots, \left(\begin{bmatrix} x_n \\ f_{WBM(x_n)} \end{bmatrix}, y_n \right) \right\}$$

that is in turn employed accordingly to generate a BBM in the form: $f_{BBM}([x, f_{WBM}(x)])$.

In this section we will test an *A-GBM* approach by including prior information extracted from the Spline WBM (f_{WBM}) (§ 4) in the forget layer of a deep recurrent NN. This architecture takes advantage of the design principles adopted in previous sections to implement the recurrent equivalent deep learning network of the model presented in § 5.1. More specifically, instead of forming different interpretations (mapping functions) for different partitions of the input space, we attempt to utilize the adaptive nature of LSTM networks to automate the translation of different operational states $(V, \bar{V_N})$ (fluctuations) to the appropriate RPM domain. In this LSTM alternative, the time-series of this extended input vector are fed to a respective LSTM architecture, as depicted in Fig. 8.

The main extension in comparison to the architecture described in Section §5.1, is that the first layer is replaced with a layer of LSTM cells. The rest of the architecture remains the same as described in §5.1. The output h_k of the LSTM hidden layer is an embedding representing a N-dimensional space. The embedding is then connected sequentially to a standard dense layer of reduced dimensionality, k, which in turn is connected to the output layer, providing the final estimate.

Consequently, the training dataset is transformed using a moving window of size N in order to produce the respective input samples for each time step. With this transformation the LSTM network *looks back* N time steps to form the hidden state units h_{t-N}. The hidden state acts as the model's memory as it holds information on data the network has *seen* before.

In order to facilitate the construction of appropriate LSTM embeddings we introduce a new method of exploiting the information extracted by Spline interpolation (see § 4), and more specifically the generated knots of the Spline interpolant to introduce a Spline-Informed embedding space.

The main difference from traditional LSTM networks, here referred to as BaseLSTM, is that we replace the affine transformations (e.g: $\mathbf{y} = \mathbf{x} * \mathbb{A}^T + \mathbf{b}$) in the **Forget Module**(f_t) of the network with the cubic Spline representations of the **current** (x_t) and the **hidden states** (h_{t-N}) to traverse from the traditional forget gate representation (Eq. (10)a) to the representation described by equation (10 b).

$$f_t = \sigma(W_f * [h_{t-N}, x_t] + b_f) \quad (a) \quad \longrightarrow \quad f_t = \sigma(f_{WBM}([h_{t-N}, x_t]) \quad (b) \tag{10}$$

By altering the forget layer of the LSTM network, we are able to induce prior information extracted from the **Spline WBM** (f_{WBM}) into the LSTM architecture and construct, as proven next, "meaningful" embedding representations of the data by mapping them into a physics informed space. The internal topology of the *SplineLSTM* network is depicted below in Fig. 9. Fig. 10 provides a preliminary comparison of the performance of *SplineLSTM*, as an alternative grey box approach, versus that of the traditional $BaseLSTM$ for

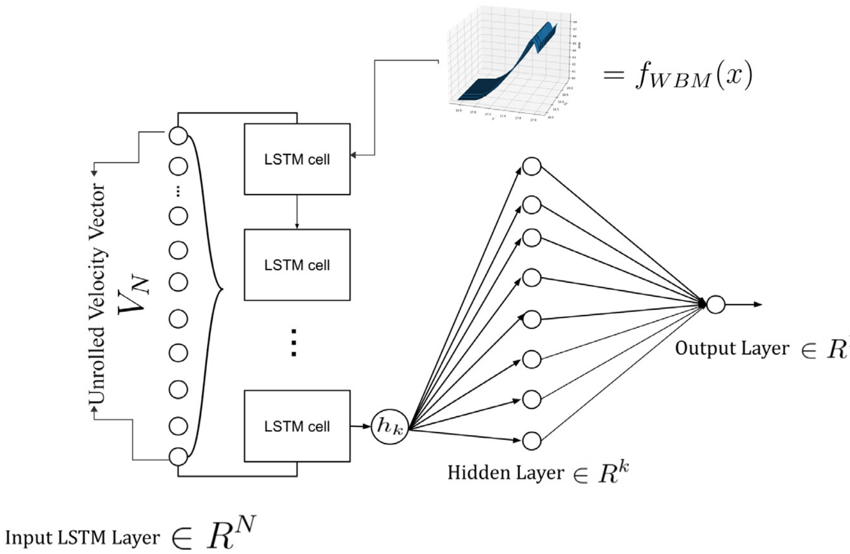


Fig. 8. The *SplineLSTM* architecture.

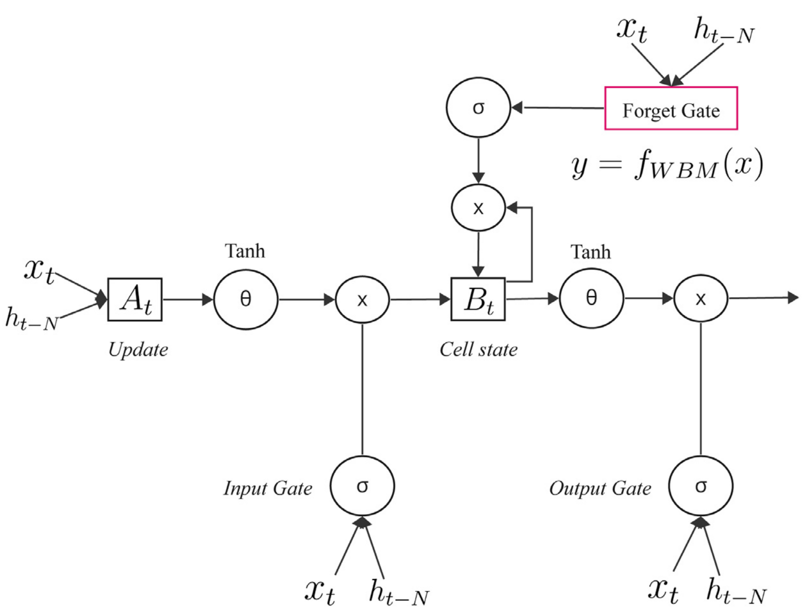


Fig. 9. *SplineLSTM* internal network topology.

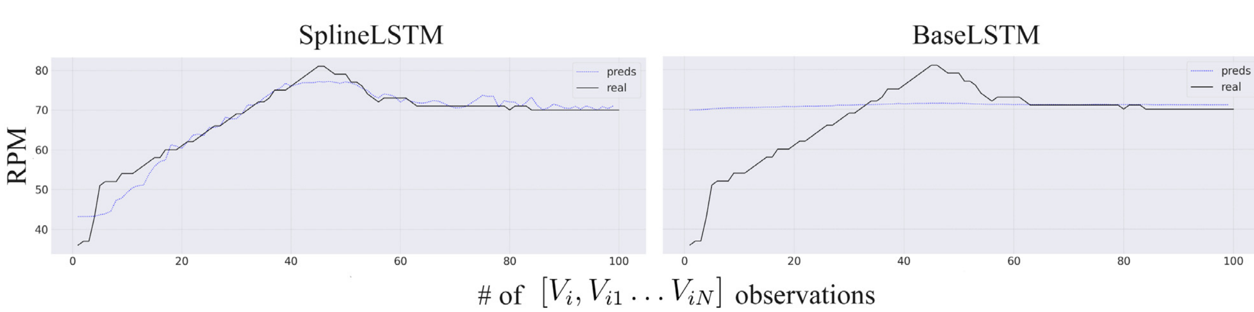


Fig. 10. Performance comparison between *SplineLSTM* and $BaseLSTM$.

a set of $\approx$ 100 $[V_i, V_{i1} \dots V_{iN}]$ observations. A more detailed experimental evaluation of the *SplineLSTM* network will be conducted in § 6.

At this point we conclude with the list of proposed alternatives for tackling the RPM estimation challenge. In summary, we have developed a number of approaches based on the following concepts:

- Utilize DTri (Delaunay Triangulation), or K-Means to find neighborhoods of similar $(V, \bar{V_N})$ fluctuations and construct piecewise Cubic B-spline interpolants on different clusters in order to approximate RPM; see in § 4.
- Utilize an ensemble of local models, with learnt contributions in RPM estimation, see in § 5.1.

Table 1
RPM Prediction & Clustering techniques summary.

Type	Technique	Abbreviation	Section
Prediction	Linear Regression	LR	4.1
	Spline Regression	SR	4.1
	Random Forest Regression	RF	4.1
	Weighted average Neural	*wAvgNN*	5.1
	LSTM with velocity time-series	*BaseLSTM*	5.2
	LSTM with extended input (multiple) time-series	*SplineLSTM*	5.2.1
Clustering	K-Means	KM	4.1
	Delaunay Triangulation based clustering	DTC	4.1

- Utilize LSTM architectures, taking into account a window of the original time-series instead of an average of the values in that window; see in §§ 5.2 and 5.2.1.

6. Experimental evaluation

In this section we perform a series of experiments on multiple (Velocity, RPM) time-series datasets, to evaluate the performance of the different methods presented in § 5. The aim is to interpret the performance of our estimators with respect to space partitioning, splines, and Deep Learning modeling, and compare them with popular baseline regression techniques (e.g. Linear and Random Forest Regression) and the analytical method described in § 4.

6.1. Methods summary

Table 1 collects the various RPM prediction techniques by providing their complete name and abbreviation, as well as a pointer to the section where they are defined in this text. It also refers to the two clustering techniques that we introduced in § 4.1.

6.2. Dataset

All the experiments were conducted on real data, using a dataset comprising Velocity (Speed overground) and RPM values from two existing container ship vessels[5] with a carrying capacity of 10,000 and 8000 TEU's.[6] The values were acquired by exploiting the streaming and orchestration capabilities of the DT framework proposed in § 3 and correspond to a vast majority of different round-trip voyages at different periods and geographical locations. As a whole, the dataset covers a time span of one year for both vessels, (March 2019 - March 2020) with approximately 400,000 data points. In order to examine the statistical significance of our results, we created 10 statistically independent subsets, for each vessel, extracted from different time periods of approximately 5,000 data points, that cover 84 hours or 3.5 days of the vessel's trip.

In Fig. 11 we visualize the majority of the round trip voyages conducted by the two container ship vessels during one year, as well as a snapshot for one route visualizing the corresponding weather at a specific time and location during the vessel's voyage.

From these datasets, the 80% was used for training and the rest for testing. Statistical independence was preserved between different datasets with the use of the Kolmogorov-Smirnov test (KS-test). This is a two-sided test for the null hypothesis that 2 independent samples are drawn from the same continuous distribution. The dataset used in the context of this work is available, in sanitized form, upon request to the first of authors. It consists of approximately $5 * 10^4$ observations of V, RPM, and their corresponding timestamp.

6.2.1. Data cleaning

The raw data, collected from the sensors of the vessel, are in time-series (minutely) form and tend to be "noisy" and even erroneous in some cases. In order to remove noise, we employed a fit & filter technique that effectively "cleaned" the data but at the same time kept the bulk of information needed for training robust prediction models. Data filtering was implemented in two stages. First, assuming that our dataset follows a normal-like distribution, we filter out data points that are outside of the 95% confidence interval and keep values for each feature that lie within a 2-times standard-deviation band from the mean value. Then we transform our dataset into 15-min rolling window averages in order to further smoothen out any spikes and outliers that occur in the feature set from the sensors onboard ship. The raw vessel's speed and corresponding RPM collected from the sensors versus the mean values and 15 min rolling window averages are depicted in Fig. 12.

6.3. Comparative analysis

In this section we will demonstrate experimental results obtained by utilizing the methods introduced in previous sections. At the end of this section we will be able to draw significant statistically conclusions versus the following issues:

✓ How the WBM of § 4, combined with either KM or DTC, performs in comparison with the rest of the ML and DL techniques proposed.
✓ Best combination of clustering technique and ML regression method.
✓ Overall best method to utilize for RPM approximation.

In order to comparatively evaluate the different approximation techniques, we cross validate the performance of our estimators in all subsets by calculating the Mean Absolute Error (MAE) in each one of them as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{11}$$

where y_i is the prediction, x_i the true value and $n+1$ is the size of the dataset.

Then we compute the average MAE across all subsets. Table 2[7] reports the results for all techniques, showing the average MAE values at the 95% confidence interval. When a prediction method is combined with a clustering technique, MAE is reported for the best clustering scheme. Since the best number of clusters may vary across subsets, we report the average number of clusters across the datasets. In the experiments conducted for K-Means, the estimator's performance was tested on a fixed range of clusters $(1-11)$, whereas for Delaunay Triangulation clustering the estimator's performance was tested on a range of cut-off values from 0.2 to 1.4 that generates an arbitrary number of clusters on each run, varying from 1 to 19 clusters.

The results in Table 2 show that the methods, which combine the local models using weighted average (*wAvgNN*), tend to perform better than the baseline ML regression models like LR, RF as well as SR. Also,

[5] In the next paragraphs we will refer to data and results corresponding to the first and second container-ship with the $\mathbf{CS}_1$ and $\mathbf{CS}_2$ abbreviation respectively

[6] (Twenty-foot Equivalent Unit - unit of cargo capacity used for container ships. and terminals)

[7] In Table 2 we showcase the best method with bold, the second best underlined and all statistically significant results appear with a star symbol.

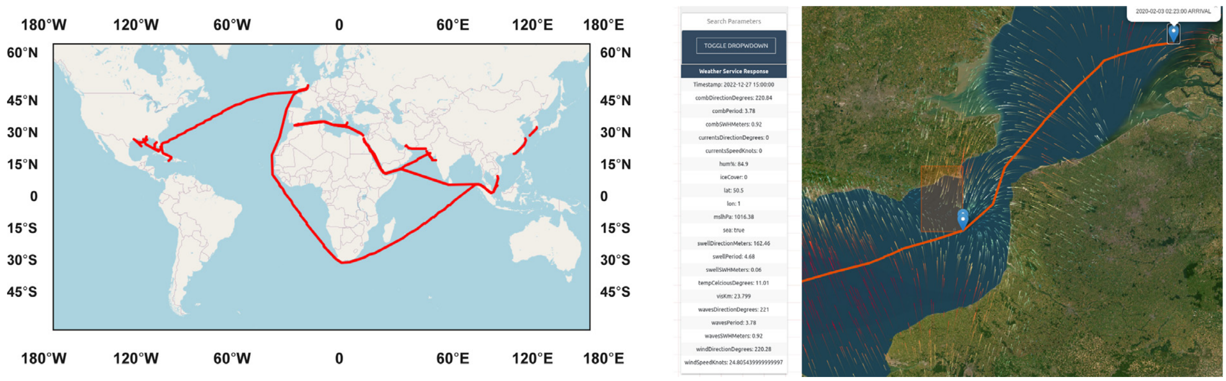


Fig. 11. Routes visualization.

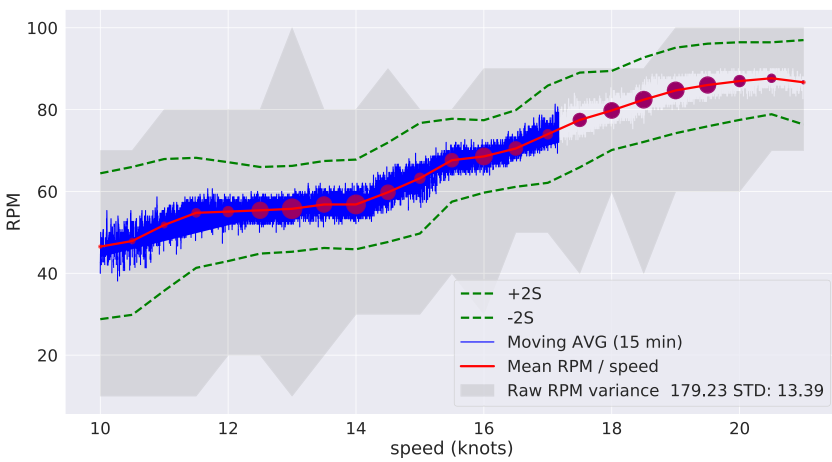


Fig. 12. RAW data values VS Mean values VS Rolling window average values.

Table 2
Average MAE & standard error in RPM values prediction for the different regression methods (smaller values are better) and the optimal number of clusters.

	Mean Absolute Error (average)			Number of clusters (average)	
	K-Means	DTC	No clustering	K-Means	DTC
LR	3.14 ± 0.1	3.2 ± 0.02	3.2 ± 0.01	5.34	19.30
SR	2.76 ± 0.14	2.12 ± 0.56*	2.77 ± 0.093	6.60	8.23
RF	3.33 ± 0.078	3.67 ± 0.036	2.36 ± 0.05*	5.67	4.20
wAvgNN	2.43 ± 0.023	2.15 ±0.31*	5.38 ± 1.84	6.80	21.30
BaseLSTM	-	-	4.35 ± 0.434	-	-
SplineLSTM	-	-	**1.73 ± 0.477***	-	-

there are certain estimators such as Random Forest (RF) that tend to perform better without clustering. Another conclusion we can extract from the above table is that Spline regression (SR) exhibits promising performance. Especially, SR coupled with DTC (referred in next paragraphs as SR_{DTC}) results in much higher accuracy than any of the other methods when coupled with DTC. This result is aligned with pertinent literature that states a connection between Delaunay triangulation and splines (Musin, 1997) and with our findings in § 4, where we explored the approximation power of piecewise estimators constructed on arbitrary input space partitions to estimate RPM. Furthermore, overall *SplineLSTM*, seems to have the lowest MAE overall with *wAvgNN* coming second and then SR. Results and conclusions were derived after detailed evaluation of all the methods on 10 different subsets of approximately 5000 observations. Statistical significance of the results was preserved, by utilizing paired *T-test* between the lists of errors for the different methodologies proposed.

Taking into account the results depicted in Table 2, we visualize, in Fig. 13, the results for the two different clustering techniques for the weighted average NN (*wAvgNN*) and for the baseline regression methods SR, LR, RF. We can clearly observe that DTC works significantly better when coupled with methods that used Spline approximation either directly as in the WBM model (SR) or as in the weighted average deep learning implementation *wAvgNN*. From the plots depicted in Fig. 13, we can conclude that there exists no trend between the number of clusters and the performance of the estimators that implies a certain monotonic relationship (either positive or negative) among them. This claim is also supported by conducting statistical significance evaluation (Spearman correlation test between the number of clusters and the average performance of our estimators (MAE)), for 5 statistically independent subsets for both K-Means and DTC. The Spearman correlation results are as follows:

- K-means: correlation $\simeq 0.05$ & p-value $\simeq 0.74$
- DTC: correlation $\simeq 0.12$ & p-value $\simeq 0.34$

In Fig. 14 we depict the overall performance of the compared models, on 10 statistically independent subsets. We combined every model

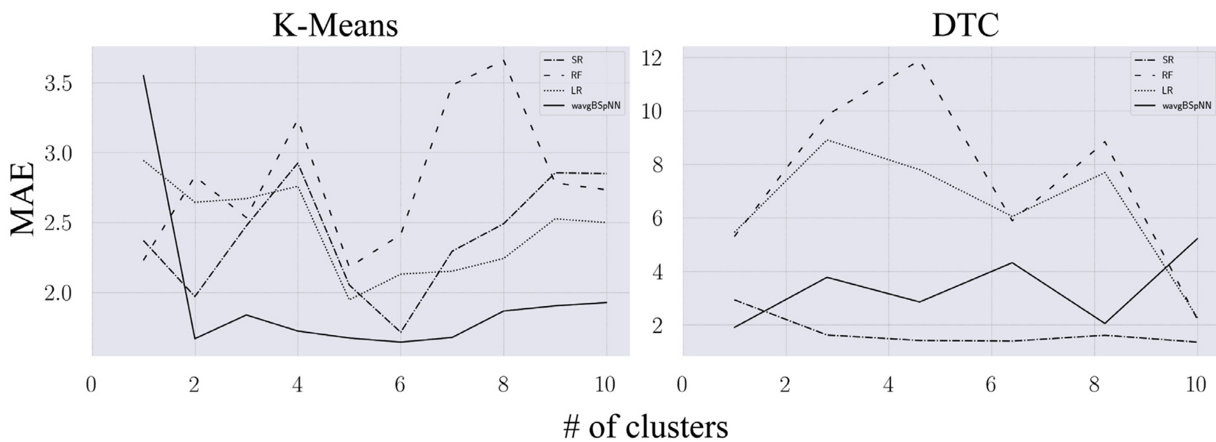


Fig. 13. Models performance comparison through clusters.

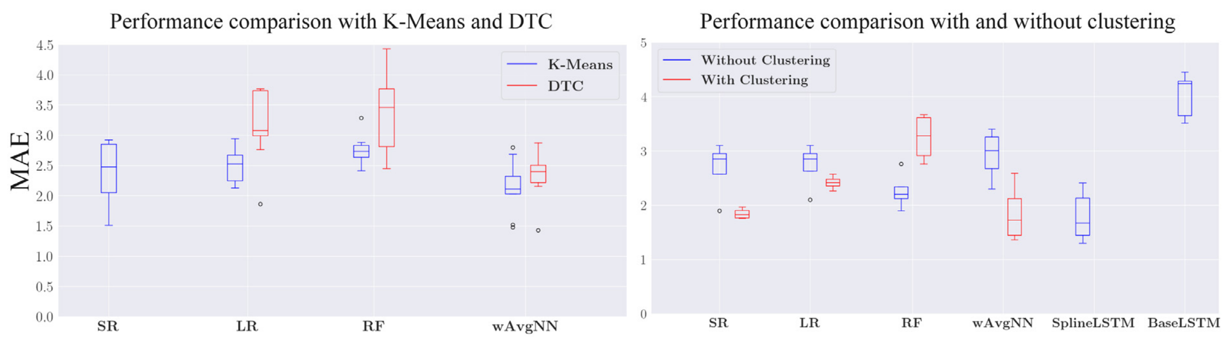


Fig. 14. Average performance comparison for estimators .

with either K-means or DTri clustering and compute the Mean Absolute Error (MAE) between the actual and predicted RPM values for each subset. Finally, we plot the distribution of average MAE values in all clusters of each clustering scheme for all datasets. In order to compare the effect of clustering, in the same figure we also visualize the distribution of average MAE values in each subset, either by cluster or in the subset as a whole. When clustering is applied the optimal number of clusters in terms of accuracy for either DTC or KM is selected across all 10 different subsets. As we can readily observe from Fig. 14 in all methods, apart from RF, when clustering is applied, the performance of the respective model outperforms that of the same model when "trained" in the dataset as a whole (lower MAE values).

Visualizing performance of the most prominent methods

In the following, we showcase a snapshot of the average performance for the top three estimators as presented in the experimental evaluation section above, across a test set of $\simeq$ 1000 observations. The estimators were trained on a dataset of approximately $5 * 10^3$ observations representative of the most common states of a vessel during a voyage. For *wAvgNN* we achieved accuracy 95.86% with an optimal number of 7 clusters whereas for SR_{DTC} we achieved 92.41% accuracy with an optimal number of 8 clusters. *SplineLSTM* achieved 97.68% accuracy on this particular test set. In Fig. 15 we provide an illustrative example of the actual performance of our estimators on a randomly picked voyage of 7 days from the test set to visualize the accuracy of our estimators on different speed ranges. The estimates are extremely close to the actual RPM values but ultimately *SplineLSTM* is able to infer RPM values in a more accurate manner than the other two methods. The bars on the graph indicate the number of observations found for a particular speed range ($\pm$ 0.5).

6.4. Utilizing RPM predictions and vessel specifics to approximate FOC

This section illustrates how a FOC estimate can be made on the basis of the input (velocity measurements) and output (RPM) of the proposed method, in conjunction with basic vessel's particulars, and standard Marine-Engineering knowledge. The experiments that compare the FOC estimations with actual FOC data of a container ship for different voyages, demonstrate the potential of the proposed method.

In summary, the proposed FOC estimate can be written as:

$$FOC = 2\pi\rho \cdot SFOC \cdot K_Q(J) \cdot \left(RPM/60\right)^3 \cdot D^5, \quad (12)$$

where:

- ρ is the sea water density ($\approx 1026kg/m^3$),
- $SFOC$ is the *Specific Fuel Oil Consumption* of the Main Engine per energy unit, which is available by the ship owner,
- D is the diameter of the propeller, and
- $K_Q(J)$ is the torque coefficient, depending on the so-called *advance coefficient* J, further analyzed in the sequel.

If Q is the torque delivered by the Main Engine on the propeller shaft, then for the Wageningen B-series of propellers (Oosterveld & van Oossanen, 1975), which are widely used by the Marine-Engineering practitioners, regression analysis of a large volume of experimental data produced by the Netherlands Ship Model Basin (NSMB) in Wageningen via open-water experiments with 120 propeller models, led to the following regression formula:

$$Q = K_Q(J)n^2 \cdot D^5, \qquad n = RPM/60. \quad (13)$$

Here $K_Q(J)$ is a polynomial function of J (Bernitsas, Ray, & Kinley, 1981) provided some further propeller particulars are known, such as the pitch over diameter ratio P/D, the number z of propeller blades and the blade-area ratio A_E/A_0.

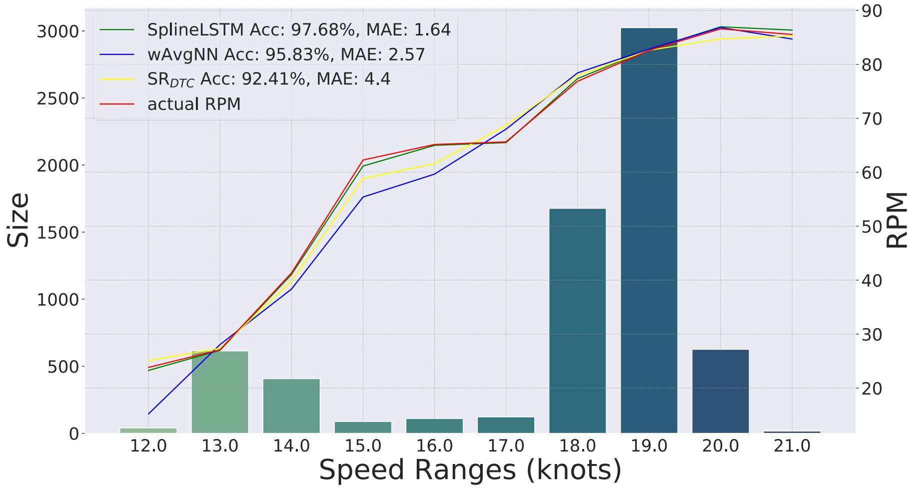


Fig. 15. Performance comparison between *SplineLSTM* - *wAvgNN* - SR$_{DTC}$.

Table 3
Computational performance of the FOC-estimate formula (12).

	Total Act FOC(MT)	Total Pred FOC(MT)	FOC Abs Diff(MT)	FOC Perc Diff
[**CS**$_1$]: SUEZ - ROTTERDAM	1100.62	1104.59	3.97	0.35
[**CS**$_1$]: TANGER MED - SUEZ	411.7	413.42	1.72	0.41
[**CS**$_2$]: TAMPA - TANGER MED	802.01	804.41	2.4	0.24
[**CS**$_2$]: PUSAN - PANAMA	489.27	490.86	1.59	0.32
Total	**2812.62**	**2813.28**	**9.68**	**1.32**

The primary parameter for $K_Q(J)$ is the so-called advance coefficient

$$J = \frac{V_A}{nD}, \tag{14}$$

where V_A is speed of advance of the propeller relative to the water in which it is working, which is lower than the speed of the vessel V. This is expressed by

$$V_A = V(1 - w), \tag{15}$$

where w is referred to as the wake fraction coefficient. For ships with one propeller, as it in our test case, w is normally in the range of 0.20 to 0.45. Furthermore, since containerships do not have large block coefficient, we have used values w in $[0.25, 0.30]$ reflecting the fact that the distribution of the water velocity around the propeller will not be strongly non-homogeneous.

On the basis of the above discussion we conclude that:

1. Monitoring the velocity V and estimating RPM by employing the techniques (*SplineLSTM*) proposed in the previous sections, we have a good overview of the fluctuation of the advance coefficient J, which constitutes a key indicator of the vessel's propulsion system performance.
2. Using basic propeller particulars and the polynomial regression results provided by Bernitsas et al. (1981), formula (12) enables an easy and robust estimation of FOC.
    - To validate this assertion, we provide in Fig. 16 and Table 3 the actual and predicted average FOC for the two container-ship vessels (**CS**$_1$, **CS**$_2$) as calculated by our method, after adjusting transitional speeds (acceleration, deceleration) for four different voyages that were not included in the training set. Red circles indicate number of observations for a specific range of velocity ($V_i \pm 0.25$) during a voyage.

7. Contributions to literature & practical implications

In the context of this work, we demonstrated a prototype of a *Digital Twin-(ing) MIS* for the maritime sector that transcends beyond State-Of-The-Art (SOTA) solutions by employing a multifaceted ecosystem for operational optimization. We discussed the proliferation and importance of digitization in the waterborne industry and argued how traditional MIS can be used as the foundation for an enhanced, versatile ecosystem to achieve carbon-neutral ship operations. To this end, we extracted important requirements by acquiring real word data corresponding to two different Living Labs (LLs; container vessels, utilized in § 6.4), in order to fully comprehend and assimilate the broad range of specifications required, which outlined the main components of the Operational Optimization digital suite. This adaptive ecosystem of algorithms, models, and data sources aims to vastly automate the decision-making procedure and offer the ability to shipowners and external vendors to select tailor-made mitigation strategies, depending on their own set of needs and long terms goals.

A thorough literature review and market analysis concerning existing frameworks (MIS) for operational optimization (§ 2) allowed us to identify the gaps and highlight the added value of the envisaged digital platform, showing the way beyond SOTA solutions. Requirements elicitation from the two LLs established the cornerstone of the platform and defined available resources in terms of infrastructure for data collection and storing, as well as regarding model employment and inference. By operating the two LLs on a day-to-day basis, an appropriate roadmap was established defining for each vessel the short and long-term goals (emission reduction, power management, charter party compliance) as well as the most suitable, in terms of financial and technical viability, mitigation solution. To this end, a global, cross-vessel, use case was defined; that of Emission Control and more specifically FOC approximation (§ 2). Utilizing the aforementioned prominent working example, we outlined the backbone to realize the first version of the Digital Twin MIS adapted to this specific use case.

The technical criteria and infrastructure in terms of services, frameworks, and appropriate models were defined in alignment with the aforementioned use case and an adaptive predictive module for FOC estimation was demonstrated (§ 5.2.1). In more detail, we put a strong emphasis on the theoretical aspect of the entailed work, by demonstrating an easy-to-deploy, in terms of feature acquisition, *Grey Box Method*

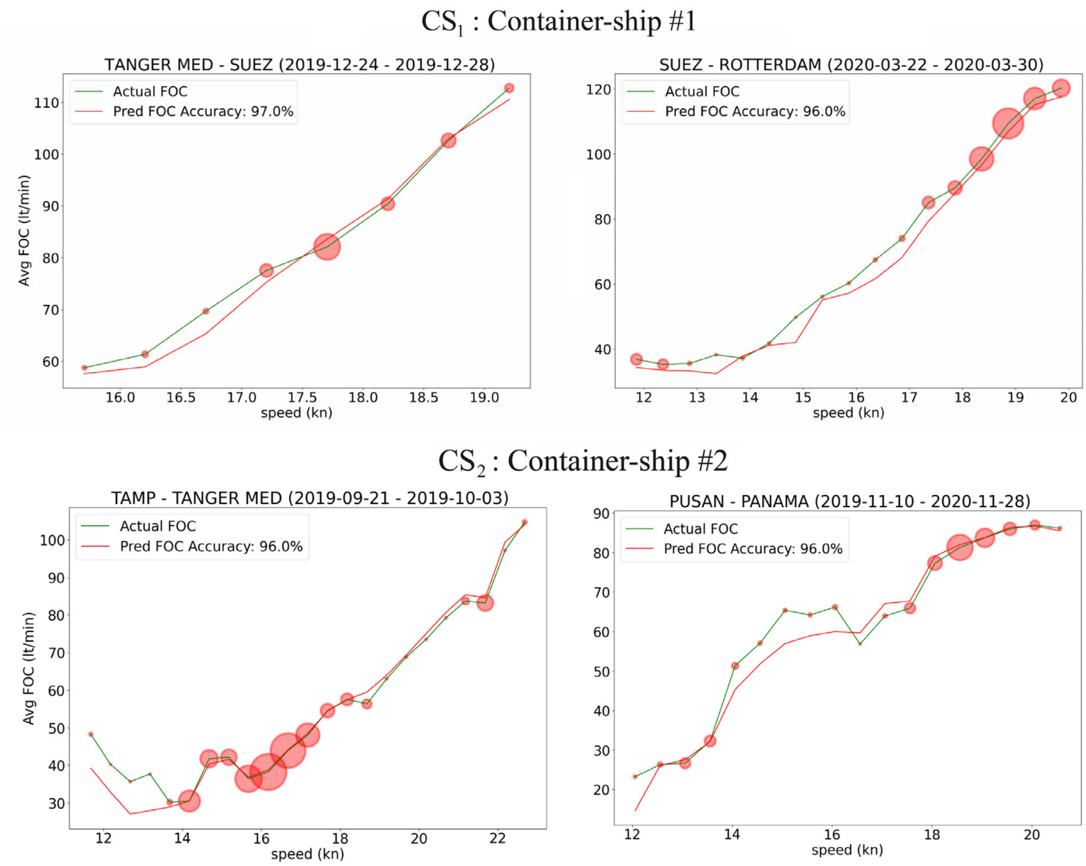


Fig. 16. Actual/Predicted FOC(lt/min) comparison in different voyages .

(GBM) for FOC estimation (§§ 4–5). This approach complements the computational power and approximation capabilities of DL (Deep Learning) models with standard marine engineering theory to offer a novel adaptive predictive scheme for emission control and regulatory adherence.

This holistic versatile platform, in conjunction with the GBM proposed, realizes a multi-tenant architecture enabling cross-vessel, model deployment, and actuation by establishing a real-time vessel-to-shore and vessel-to-vessel communication. Immediate planning and next steps were also defined, including the implementation of a Decision Support System (DSS) intercommunicating with the platform to evaluate possible solutions via a user-defined approach.

8. Conclusions and future work

The motivating real-world problem of this work is the optimization of vessel routing, described in the form of geospatial data, through minimization of its FOC (Fuel-Oil Consumption). Via reviewing the pertinent literature on the subject and conducting initial experimentation we concluded that the problem could be handled efficiently, if a robust model for predicting the RPM (revolutionary speed of the main engine) of a vessel, moving with known speed V, were available. Furthermore, access to real industrial data obtained from measurements on-board ships, indicates a strong linear correlation between RPM and V on specific time intervals during a voyage. However, non-linearities do also emerge during other intervals.

On the basis of the above, we have been led to the idea of developing an RPM predictive model that decomposes the domain in correlated sub-domains with respect to velocity V (§§ 4). In this connection, we opted for *Spline Regression (SR)* in order to approximate the underlying function $RPM(V)$ on each subdomain, as splines are by their nature continuous piecewise polynomials appropriate for approximating functions with varying behavior in partitioned domains. We also examined *Linear regression (LR), Random Forest (RF)* and a baseline *Neural Network (NN)*.

In order to combine splines with Deep Learning, we conducted an exploratory analysis by building two different Neural-Network (NN) architectures, that utilize spline regression in two different ways. The first architecture employs the deep learning equivalent of the WBM (White Box Model) described in § 4 by consolidating different estimators trained on different partitions of the input set, in a weighted average predictive scheme. Lastly, we implemented two different LSTM-based (Long short-term memory) NN architectures: one baseline LSTM and another one that introduces a Spline-informed embedding space based on the psychics-informed WBM model.

To the best of our knowledge it is the first time that FOC approximation is addressed, under the prism of a fully automated digitized MIS, as the one proposed in the context of this work. Furthermore the methodology employed, leverages the approximation capabilities of ensemble deep learning, with standard marine engineering theory to conclude to a consolidated, physics-informed, FOC predictive scheme. The reduced and easily acquired feature set constitutes another novelty, as it automatically renders the proposed methodology, a generic, applicable to all vessels approach. Conventional methods found in pertinent literature concerning FOC estimation are vessel and/or data dependent, and lack the theoretical support and bench-marking criteria of methods that exploit and consolidate the backbone of AI approximation theory and naval engineering. Finally, the envisaged framework as a whole, constitutes a pivotal and cross sectoral venture that aims to exploit data from different stakeholders attached to the waterborne sector and enable sensing and control actuation on the vessel, by evaluating the proposed methodology in a real world setting.

Summarizing the obtained results, we see that most of the methods, trained on sub-spaces (clusters) and created through clustering (k-means or DT-based algorithms), achieve higher accuracy compared to a generic LR or NN model. Especially, when we combine the approximation power of all local models with the global model on a weighted average predictive scheme, we are able to achieve higher accuracy than simply averaging the most suitable local estimator with the global model. Furthermore, combining the triangulation with piecewise regression, showed promising results.

Besides expanding the scale and variability of our experiments with new datasets, our short-term objective will focus on investigating the effect of hyper-parameters and options (including distance metric for the clustering), as follows:

- the optimal cut-off value for the DT-based clustering algorithm and generally the optimal number of clusters for either of the two proposed clustering methods;
- the distance metric, since so far only Euclidean distance has been tested;
- the population and the appropriate placement of the knots used to approximate the underlying function on each partition;

Another important direction for our research is to address the issue of 'lack of historical data' in the maritime industry by employing a **'knowledge sharing' platform across vessels**. The main approach will be to exploit the *Digital -Twining* framework proposed in the context of this work, to train our proposed methods in a continuous setting (incremental learning) for different types of fleets (Containers-Tankers-LNG-RoRo) utilizing as input an expanded feature set consisting of weather features and a variety of vessel-specific parameters(propeller and hull geometry). This generic model will be adjusted initially based on these vessel specifics features in order to provide accurate FOC predictions for ships of the same type that are missing data (transfer learning). In a latter stage the model will be further evaluated and refined, *ON EDGE*, by utilizing another core functionality of the DT framework (*Control Actuation layer*). With this methodology we aim to assemble a *'Vessels Specific Induced, FOC Predictive Library'* of generic, physics-informed models, for different types of vessels. The core functionality of this generic, physics-informed cross vessel library conceptualizes the emerging concept of the so called **Digital Twin** in the shipping industry, as described in the context of this work.

Author Agreement Statement

We the undersigned declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We understand that the Corresponding Author is the sole contact for the Editorial process. He is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs.

Signed by all authors as follows: Dimitrios Kaklis, George Giannakopoulos, Iraklis Varlamis, Constantine D. Spyropoulos, Takis J. Varelas

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Dimitrios Kaklis: Conceptualization, Methodology, Software. **Iraklis Varlamis:** Conceptualization, Methodology, Software. **George Giannakopoulos:** Conceptualization, Methodology. **Takis J. Varelas:** Data curation. **Constantine D. Spyropoulos:** Conceptualization, Methodology.

Acknowledgments

This Publication was supported partially by the program of Industrial Scholarships of Stavros Niarchos Foundation, and partially by the DT4GS project, a European Union's Horizon 2020 research and Innovation programme under the Marie Skłodowska-Curie Grant Agreement No 874990. Access to industrial data has been provided by Danaos Shipping Co.

References

Ahlgren Fredrik, M. E. M., & Thern, M. (2018). Predicting dynamic fuel oil consumption on ships with automated machine learning. *Journal of Mechanical Science and Technology, 32*(12), 5785–5796.

Avgouleas, K. (2008). Optimal ship routing. MSC in Mechanical Engineering, MIT.

Bai, Z., Shangguan, W., Cai, B., & Chai, L. (2019, July). Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic. In 2019 Chinese Control Conference (CCC) (pp. 8600–8605). IEEE.

Bernitsas, M., Ray, D., & Kinley, P. (1981). K_T, K_Q and efficiency curves for the Wageningen B-series propellers. *Department of Naval Architecture and Marine Engineering College of Engineering*. The University of Michigan.

Bohlin, T. (1992). *Interactive system identification: Prospects and pitfalls*. Springer-Verlag.

Campolucci, P., Capperelli, F., Guarnieri, S., Piazza, F., & Uncini, A. (1996). Neural networks with adaptive spline activation function. In *Proceedings of 8th mediterranean electrotechnical conference on industrial applications in power systems, computer science and telecommunications (MELECON 96): vol. 3* (pp. 1442–1445). IEEE.

Chen, S., Hong, X., Khalaf, E., Morfeq, A., & Alotaibi, N. D. (2015). Adaptive B-spline neural network based nonlinear equalization for high-order QAM systems with nonlinear transmit high power amplifier. *Digital Signal Processing, 40*, 238–249.

Chen, X., Xu, X., Yang, Y., Wu, H., Tang, J., & Zhao, J. (2020). Augmented ship tracking under occlusion conditions from maritime surveillance videos. *IEEE Access, 8*, 42884–42897.

Chondrodima, E., Georgiou, H., Pelekis, N., & Theodoridis, Y. (2020). Particle swarm optimization and RBF neural networks for public transport arrival time prediction using GTFS data. *International Journal of Information Management Data Insights, 2*(6), 489–497.

Cootes, T. F., Ionita, M. C., Lindner, C., & Sauer, P. (2012). Robust and accurate shape model fitting using random forest regression voting, *In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII* (*12*, pp. 278–291). Springer Berlin Heidelberg.

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.

Deepu, T. S., & Ravi, V. (2021). Supply chain digitalization: An integrated MCDM approach for inter-organizational information systems selection in an electronic supply chain. *International Journal of Information Management Data Insights, 1*(2), 100038.

Eldershaw, C., & Hegland, M. (1997). Cluster analysis using triangulation. *Computational Techniques and Applications*, 201–208.

Fermani, M., Rossit, D. G., & Toncovich, A. (2021). A simulated annealing algorithm for solving a routing problem in the context of municipal solid waste collection. *International Conference of Production Research – Americas, 1408*(6), 63–79.

Fey, M., Eric Lenssen, J., Weichert, F., & Müller, H. (2018). SplineCNN: Fast geometric deep learning with continuous B-spline kernels. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 869–877).

Filippopoulos, I., Panagiotakopoulos, T., Skiadas, C., Triantafyllou, S.-M., Violaris, A., & Kiouvrekis, Y. (2022). Live vessels' monitoring using geographic information and internet of things. In *2022 13th international conference on information, intelligence, systems & applications (IISA)* (pp. 1–7). IEEE.

Folgheraiter, M. (2016). A combined B-spline-neural-network and ARX model for online identification of nonlinear dynamic actuation systems. *Neurocomputing, 175*, 433–442.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 1–67.

Garg, R., Kiwelekar, A. W., Netak, L. D., & Ghodake, A. (2021). i-Pulse: A NLP based novel approach for employee engagement in logistics organization. *International Journal of Information Management Data Insights, 1*(6), 489–497.

Gkerekos, C., & Lazakis, I. (2020). A novel, data-driven heuristic framework for vessel weather routing. *Ocean Engineering, 197*, 106887.

Gkerekos, C., Lazakis, I., & Theotokatos, G. (2019). Machine learning models for predicting ship main engine fuel oil consumption: A comparative study. *Ocean Engineering, 188*, 106282.

Golias, M. M., Saharidis, G. K., Boile, M., Theofanis, S., & Ierapetritou, M. G. (2009). The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics, 11*(4), 358–377.

Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. *White. paper, 1*(2014), 1–7.

Grover, P., & Kar, A. K. (2017). Big data analytics: A review on theoretical contributions and tools used in literature. *Global Journal of Flexible Systems Management, 1*(2).

Gupta, P., Taskar, B., Steen, S., & Rasheed, A. (2021). Statistical modeling of ship's hydrodynamic performance indicator. *Applied Ocean Research, 111*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2019). Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine, 57*(6), 114–119.

Jain, S., Seeja, K. R., & Jindal, R. (2021). A fuzzy ontology framework in information retrieval using semantic query expansion. *International Journal of Information Management Data Insights, 1*(1), 100009.

Jeon, M., Noh, Y., Shin, Y., Lim, O.-K., Lee, I., & Cho, D. (2018). Prediction of ship fuel consumption by using an artificial neural network. *Journal of Mechanical Science and Technology, 32*(12), 5785–5796.

Kaklis, D., Eirinakis, P., Giannakopoulos, G., Spyropoulos, C., Varelas, T. J., & Varlamis, I. (2022a). A big data approach for fuel oil consumption estimation in the maritime industry. In *2022 IEEE eighth international conference on big data computing service and applications (BigDataService)* (pp. 39–47). IEEE.

Kaklis, D., Giannakopoulos, G., Varlamis, I., Spyropoulos, C. D., & Varelas, T. J. (2019). A data mining approach for predicting main-engine rotational speed from vessel-data measurements. In *Proceedings of the 23rd international database applications & engineering symposium* (pp. 1–10).

Kaklis, D., Varlamis, I., Giannakopoulos, G., Spyropoulos, C., & Varelas, T. J. (2022b). Online training for fuel oil consumption estimation: A data driven approach. In *2022 23rd IEEE International Conference on Mobile Data Management (MDM)* (pp. 394–400). IEEE.

Kar, A. K., & Kushwaha, A. K. (2021). Facilitators and barriers of artificial intelligence adoption in business–insights from opinions using big data analytics. *Information Systems Frontiers*, 1–24.

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. *IEEE Access, 6*, 1662–1669.

Kim, B., & Kim, T.-W. (2017). Weather routing for offshore transportation using genetic algorithm. *Applied Ocean Research, 63*, 262–275.

Kouzari, E., Sotiriadis, L., & Stamelos, I. (2023). Enterprise information management systems development two cases of mining for process conformance. *International Journal of Information Management Data Insights, 3*(1), 100141.

Krata, P., & Szlapczynska, J. (2018). Ship weather routing optimization with dynamic constraints based on reliable synchronous roll prediction. *Ocean Engineering, 150*, 124–137.

Lin, J. (2012). Identification of road surface power spectrum density based on a new cubic spline weight neural network. *Energy Procedia, 17*, 534–539.

Lloyd, S. P. (1957). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 2*(12), 281–297.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1. University of California: 32* (pp. 281–297).

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. In *Proceedings: vol. 89* (pp. 89–94). Presses universitaires de Louvain.

Marc, S. (2022). Deep learning in business analytics: A clash of expectations and reality. *International Journal of Information Management Data Insights, 1*(2).

Miyeon, J., Noh, Y., Shin, Y., O-Kaung Lim, I. L., & Cho, D. (2018). Prediction of ship fuel consumption by using an artificial neural network. *Journal of Mechanical Science and Technology, 32*(12), 5785–5796.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Musin, O. R. (1997). Properties of the delaunay triangulation. In *Proceedings of the 13th annual symposium on computational geometry* (pp. 424–426).

Neamtu, M. (2007). Delaunay configurations and multivariate splines: A generalization of a result of BN Delaunay. *Transactions of the American Mathematical Society, 359*(7), 2993–3004.

Oosterveld, M. W. C., & van Oossanen, P. (1975). Further computer-analyzed data of the Wageningen B-screw series. *International shipbuilding progress, 22*(251), 251–262.

Pallotta, G., Vespe, M., & Bryan, K. (2013). Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy, 15*(6), 2218–2245.

Papandreou, C., & Ziakopoulos, A. (2020). Predicting VLCC fuel consumption with machine learning using operationally available sensor data. *Ocean Engineering, 197*, 106887.

Rippa, S. (1990). Minimal roughness property of the delaunay triangulation. *Computer Aided Geometric Design, 7*(6), 489–497.

Roh, M.-I., & Lee, K.-Y. (2018). *Computational ship design*: vol. 43. Springer.

Sarkar, B. D., & Shankar, R. (2021). Understanding the barriers of port logistics for effective operation in the industry 4.0 era: Data-driven decision making. *International Journal of Information Management Data Insights, 1*(2), 100031.

Savitha, R., Al Mamun, A., et al., (2017). Regional ocean wave height prediction using sequential learning neural networks. *Ocean Engineering, 129*, 605–612.

Telfer, E. (1963). The design presentation of ship model resistance data. *International Shipbuilding Progress, 10*(106), 215–232.

Varelas, T., Archontaki, S., Dimotikalis, J., Turan, O., Lazakis, I., & Varelas, O. (2013). Optimizing ship routing to maximize fleet revenue at Danaos. *Interfaces, 43*(1), 37–47.

Venkatachalam, P., & Ray, S. (2022). How do context-aware artificial intelligence algorithms used in fitness recommender systems? A literature review and research agenda. *International Journal of Information Management Data Insights, 2*(2), 100139.

Vettor, R., & Soares, C. G. (2015). Multi-objective route optimization for onboard decision support system. In A. Weintrit, & T. Neumann (Eds.), *Information, Communication and Environment: Marine Navigation and Safety of Sea Transportation* (pp. 99–106). Leiden, The Netherlands: CRC Press, Taylor & Francis Group.

Walsh, C., & Bows, A. (2012). Size matters: Exploring the importance of vessel characteristics to inform estimates of shipping emissions. *Applied Energy, 98*, 128–137.

Wang, L., Wang, G., Lai, M.-J., & Gao, L. (2020). Efficient estimation of partially linear models for spatial data over complex domains. *Statistica Sinica, 30*, 347–369.

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.

Wei, X., Yang, H., & Huang, W. (2021). A genetic-algorithm-based optimization routing for FANETs. *Frontiers in Neurorobotics, 15*, 697624.

Zhao, Z., Ji, K., Xing, X., Zou, H., & Zhou, S. (2014). Ship surveillance by integration of space-borne SAR and AIS-further research. *The Journal of Navigation, 67*(2), 295.

Zhengyu, L., Donald, S. R., Williams, B. W., & Xiangning, H. (2007). Online modeling for switched reluctance motors using B-spline neural networks. *IEEE Transactions on Industrial electronics, 54*(6), 3317–3322.

Zhu, Y., Zuo, Y., & Li, T. (2020). Predicting ship fuel consumption based on LSTM neural network. In *2020 7th International conference on information, cybernetics, and computational social systems (ICCSS): 32* (pp. 310–313).