A big data approach for Fuel Oil Consumption estimation in the maritime industry

Dimitrios Kaklis
Dep. of Informatics and Telematics
Harokopio University of Athens,
Danaos Research Center,
NCSR Demokritos
Athens, Greece
kaklis1992@gmail.com

Pavlos Eirinakis
Department of Industrial
Management & Technology
University of Piraeus
Piraeus, Greece
pavlose@unipi.gr

George Giannakopoulos
Institute of Informatics & Telecoms
NCSR Demokritos
Athens, Greece
ggianna@iit.demokritos.gr

Constantine Spyropoulos
Institute of Informatics & Telecoms
NCSR Demokritos
Athens, Greece
costass@iit.demokritos.gr

Takis J. Varelas
Danaos Research Center
Piraeus Greece
drc@danaos.gr

Iraklis Varlamis
Department of Informatics and Telematics
Harokopio University of Athens
Athens, Greece
varlamis@hua.gr

***Abstract*—Route optimization has been a research topic for many years in the maritime industry and it constitutes one of the key components to improving energy efficiency and sustainability in ship operations. This paper deals with the challenge of estimating Fuel Oil Consumption (FOC) in the context of Weather Routing (WR). Given a plethora of features collected from the vessel's Automatic Identification System (AIS) or on-board sensor installations, we examine how a predictive FOC scheme can be coupled with WR optimization algorithms in order to reduce the vessel's FOC, emissions, and the overall cost of a voyage. In order to handle the amount of data required for FOC prediction, we employ a streaming pipeline that harvests data in real-time from different sources and processes them appropriately for visualization, causal analysis, and forecasting purposes. In this direction, we first conduct an exploratory analysis to examine and unveil the importance and inter-association between the various variables related to sea-keeping and weather features, in order to utilize them effectively in the context of a FOC predictive scheme. Furthermore, we introduce a novel recurrent neural network architecture that approximates ideally the underlying function describing the features and the vessel's FOC by taking into account historical data, and we showcase the results. Finally, we demonstrate how the FOC prediction model can be coupled with a WR algorithm to propose the optimal route for a vessel in terms of FOC efficiency.**

***Keywords*-big data collection, information processing, vessel data, fuel oil consumption, weather routing optimization, recurrent neural networks**

I. Introduction

The task of optimal route planning is crucial for the shipping industry since it is strongly connected to the energy consumption of sea vessels. Fuel Oil Consumption (FOC) is highly affected by the speed of the vessel and the weather conditions during a voyage. There exists a wealth of spatio-temporal data related to the above problem, comprised of static vessel specifics (e.g. deadweight (carrying capacity), hull type, etc.), time-varying attributes outlining the vessel state (e.g. Revolutions or Rounds Per Minute of the Main Engine, Speed Through Water, etc.) and local conditions (e.g. related to weather and currents). The weather state constitutes the spatial dimension that affects the oversea movement cost of a ship. The respective temporal aspect relates to the environmental and ship conditions at any moment of the ship's route. Combining knowledge from both dimensions can significantly improve the FOC estimates and in turn, can lead to more informed route optimization.

FOC is closely related to the rotational speed (i.e., rounds per minute - RPM) of the main engine [1]. However, this work focuses on the relation between spatio-temporaly derived velocity measurements and weather features to employ a FOC prediction model, without *explicitly* including RPM information in the prediction. In other words, we utilize the overground velocity of a ship V, which can be easily collected by processing the $< latitude, longitude >$ coordinates of the vessel, using the Automatic Identification System (AIS) signal that is transmitted every few seconds and weather features such as wind speed, wind angle, etc., collected from a variety of services. For this purpose, we extend our previous work [1] to take advantage of a rich feature set that contains, apart from the vessel's velocity, a multitude of weather features like wind speed, relative wind angle, swell wave height, combined wave height, etc. Our work can then be effectively utilized for event recognition purposes (Post Voyage Analysis, Predictive Maintenance, etc.) and by WR algorithms aiming to reduce the emissions and maximize efficiency during a voyage. For this purpose, the work presented in this paper is structured in four parts, and demonstrates concisely the following:

- A big data management tool utilizing scheduling frameworks (*Apache Airflow*) and streaming algorithms for real-time vessel data collection, processing, and model

deployment.

- An exploratory analysis with a large-scale dataset, corresponding to a real container vessel, covering a time span of approximately one year. To extract useful features, preliminary research concerning the importance and correlations between the variables that describe the vessel's voyage is conducted. We then propose a decision-tree-based method to effectively clean the dataset from noise corresponding to faulty measurements or negligent maintenance.
- A novel Recurrent Neural Network utilizing the features extracted in the previous part of the work, to approximate the underlying function that describes the FOC of the vessel.
- Lastly, a Weather Routing (WR) optimization algorithm that integrates effectively the Recurrent Neural Network proposed for FOC estimation, in the optimization procedure.

Section II that follows analyzes pertinent literature on optimal route planning and consequently FOC estimation.

II. Related work

Optimization criteria in vessel routing include the minimization of voyage time, FOC, or voyage risk. The approaches, which have appeared so far in the literature, can be classified into four broad categories:

- *Vessel-based optimization*, which aims to optimize 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; [2, 3]
- *Environmental-based optimization*, which aims at optimizing a given route by taking into account environmental conditions, e.g., wind (speed, direction), wave (height, frequency, direction), currents; [4, 5]
- *Holistic optimization*, which combines the two previous approaches in a common context; [6, 7, 8]
- Analytical approaches trying to tackle the problem with the use of exact (NP-complete) and/or heuristic algorithms like label-setting algorithms, non-linear integer programming, or simulated annealing [9].

In order to incorporate more constraints, several methods split a vessel's voyage into areas of critical interest, involving for example zones of extreme weather conditions, emission control areas (ECAs, SECAs), high-risk zones (piracy), etc. Then, they seek for Pareto optimal solutions from a set of routes that are optimal in terms of Expected Time of Arrival (ETA), FOC, and safety or they use Genetic Algorithms [4] in order to find the best route, as a composition of optimal route segments. Methods like PSO (Particle Swarm Optimization) [10] are also employed in order to solve the multi-constraint, non-linear optimization problem of optimal route planning.

The techniques employed in the literature for estimating FOC based on vessel characteristics and/or environmental conditions can be grouped into the following categories:

- Data-oriented 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 like Support Vector Regression, Lasso Regression, and Polynomial Regression to ensemble non-parametric schemes like Random Forest (RF) regression, Decision Trees, or AdaBoost. Some studies have also experimented with baseline sequential Artificial Neural Networks (ANN) by tuning a number of hyperparameters (learning rate, number of neurons, number of layers, activation function). [11, 12].
- Approaches where machine learning (ML) methods (also known as black-box models - BBM), are combined with theoretical models (also 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 proposed models are known as grey-box models (GBM) [1, 2].

ANNs have been at the center of attention lately in many research areas. As far as vessel FOC is concerned, not many studies utilize the computational power of ANNs to approximate FOC mainly due to the problem of missing historical data. The studies found in pertinent literature dealing with FOC estimation from a deep learning perspective are presented briefly below. Some studies experiment with baseline sequential ANNs by applying a dropout in the weights in order to achieve better generalization error [5] or by tuning a number of hyperparameters (learning rate, number of neurons, number of layers, activation function) utilizing brute force methods like randomized grid search [11, 13]. In [14] a Recurrent NN is employed in order to estimate FOC but without further research as far as the architecture, or the generalization capabilities of the neural proposed.

Pertinent literature concerning applications and novelties in the maritime sector neglects the importance of a big data processing pipeline. This work deals with this omission by employing a big data management tool, continuously harvesting data related to the vessel's operational stage of life (voyage planning, cargo handling, etc). This is of high importance for ship operations and maintenance procedures, as the maritime sector has witnessed an exponential growth in data availability over the past few years. In the following, we present the workflow of a multi-purpose pipeline for data collection-processing-storing and model deployment, adapted to the needs of the maritime sector.

III. Data availability and processing

A. Features related to the prediction task

The motion of a ship through water requires energy to overcome resistance, i.e. the force working against movement. Therefore FOC is highly impacted by the total resistance of the vessel as it moves forward. Total resistance of the vessel incorporates three major components: frictional resistance, wave resistance and air resistance.

The **frictional resistance** depends on the size of the wetted area of the vessel. It represents often about 70-90% of the ship

total resistance for low-speed ships (bulk carriers and tankers), and sometimes less than 40% for high-speed ships (containers and passenger ships). **Wave Resistance** measures the effect of waves and may rise up to 30% of the total resistance. The characteristics of waves like their amplitude and wave length are determined from the ocean-wave spectra along the voyage path. Finally, **air resistance** normally represents about 2% of the total resistance, but for loaded container ships in head wind, it can be as much as 10%.

Based on the above standard marine engineering knowledge we aim to utilize meaningful features that have a prominent impact in the total resistance of the vessel like:

- Features that correspond to the frictional resistance and can be utilized in the context of a Routing Optimization algorithm, such as STW and Draft.
- Features that describe the wave resistance component, such as Wave height/Direction, Wave Period, Swell Wave Height/Direction, and Swell Period.
- Features that model the air resistance component, such as Wind Speed/Direction, Combined Wind Wave Height/Direction, and Current Speed/Direction.

All mentions to weather state variables corresponding to the direction of the feature (Wind Direction, Swell Direction, Wave Direction etc.), have been converted to relative direction taking into account the heading of the vessel.

Table I depicts a detailed representation of the feature set collected for FOC estimation purposes for each vessel and their abbreviations.

Table I: The features of our dataset

id	Feature	Abbreviation	Measurement Unit
1	Speed Through Water	STW	knots
2	Vessel Heading	VSL_H	° (degrees)
3	Mid Draft	DRFT	m
4	Wind Speed	WS	m/s
5	Wind Direction	WD	°
6	Swell Wave Height	SWH	m
7	Swell Wave Direction	SWD	°
8	Current Speed	CS	m
9	Current Direction	CD	°
10	Visibility	VIS	m
11	Temperature	TEMP	°C
12	Swell Period	SWP	sec
13	Wave Period	WP	sec
14	Combined Wave Height	CWH	m
15	Combined Wave Direction	CWD	°
16	Mean Sea Level	MSL	
17	Wave Effect	WAV_E	
18	Fuel Oil Consumption	FOC	lt/min
19	Rounds/Minute of the Main Engine	RPM_{ME}	
20	Power	P	kgw/h

B. Data collection / pre-processing pipeline

A high level visualisation of the processing pipeline is depicted in Figure 1 and comprises several steps from data collection and filtering to model building, evaluation, selection, deployment and consequently the integration of the FOC model in the WR algorithm.

Data are continuously collected from different sources (AIS, Sensors, Noon Reports, Weather Service API's) via a state-of-the-art scheduling framework (*Apache Airflow*). The pipeline harvests more than 100gb of data on a monthly basis, corresponding to routes of different vessels, which are described by the aforementioned variables.

This framework is utilized with the aim to build a fault-tolerant, modular, and multi-purpose big data tool for the maritime industry that is able to harvest data from different sources and perform tasks such as Event Recognition, Causal Analysis, Forecasting, and Incremental Training. In the scope of this work, the framework is adapted accordingly to the task of FOC training and estimation.

In the first steps, the framework integrates streaming algorithms, in *Apache Kafka* and *Spark*, that optimize data collection, processing, and storing. More specifically, the batch streaming process is handled by Kafka Cluster, which allows to balance the load of harvesting data streams in real-time from AIS and on-board monitoring systems. In continuance, the data are processed by exploiting the parallelization capabilities of Apache Spark and are eventually stored in a centralized cloud-based platform. The cleansed version of the data is consumed by a variety of data-driven models that are trained on an ideal feature set for the specified task (FOC estimation in this case), which has been extracted in the previous step. After training is complete, each model's artifacts (hyper-parameters, training error, evaluation error, convergence plots, size of dataset) are automatically logged in a web-based micro-service (*MLFlow*) to be easily accessible and comparable in order to query the most accurate model in terms of validation error. The selected model is wrapped as a web API service and is queried for inference in real-time from external applications, which in the scope of this work, is the routing optimization algorithm. After selecting the appropriate FOC prediction model, new data streams (i.e., from sensors, AIS) that are pushed to a Kafka topic on a weekly basis, are fetched once a week from the topic and used to update the model. The architecture of this pipeline gives us the advantage to leverage the streaming capabilities of Kafka, the task automation power of Airflow, and the logging features of MLFlow — all structured and orchestrated by a set of Docker containers.

The output of the pipeline can also be utilized to calculate the Energy Efficiency Operational Index (EEOI), an indicator that enables maritime industries to monitor the carbon emissions of their fleets during a voyage. EEOI is the total carbon emissions in a given time period per unit of revenue tonne-miles. The mass flow rates of CO_2, NO_x and SO_x are calculated based on the engine Power (P - kW), the Specific Fuel Oil Consumption (SFOC - $^{gr}/_{kWhr}$), accounting for the combustion stoichiometry and NO_x chemistry as follows:

$$m = P * SFOC * EF = FOC * EF \tag{1}$$

where m is the emissions mass flow rate (in $^{grams}/_{hour}$), P is the engine power, SFOC is the Specific Fuel Oil Consumption and EF is the Emissions Factor presented in Table II.

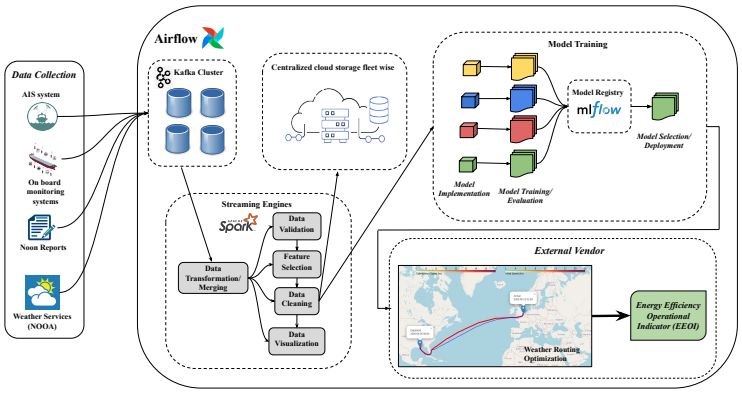


Figure 1: The pipeline from data collection to WR optimization

CO_2	3.114 (tn CO_2/ tn fuel)
NO_x	0.092 (tn NO_x / tn fuel)
SO_x	2.023 x S mass fuel fraction in fuel (tn SO_x/ tn S in fuel)

Table II: Emissions factor (EF) used for calculating CO_2, NO_x and SO_x

C. Feature selection

In order to unveil the relationships between the independent variables as well as their importance and role in estimating FOC, we conduct an initial exploratory analysis with Random Forest regression as the feature ranking algorithm. Then calculate the correlations between the most important features, and conclude to an ideal feature set that consists of independent variables that will be utilized accordingly in the context of FOC approximation.

Decision Trees (DT) is a popular classification or regression algorithm that takes into account the importance of features. More specifically, the feature importance defines the order in which features are selected for splitting the initial set of samples to subsets, from the tree root to the leafs. It is defined by the decrease in (tree) node impurity, which is weighted by the node probability. This probability is the number of samples that reach the node, divided by the total number of samples. Higher decreases in impurity denote more important features. Assuming only two child nodes (left, right) for each node, the node importance is given by the following equation:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{left(j)} C_{right(j)} \quad (2)$$

where ni_j is the importance of node j for feature i, w_j is the weighted number of samples reaching node j and C_j is the impurity of node j. Impurity is measured using Gini Index or Entropy.

The Random Forest (RF) algorithm extends the concept of Decision Trees, for high-dimensional data, by constructing many individual decision trees during training, using each time a different random subset of the initial set of features. It then collectively examines the predictions of trees in order to make the final prediction. Respectively, RF can be used to evaluate the importance of each feature across all the trees and provide a more comprehensive ranking of feature importance.

In Table III we depict the experimental results from conducting regression analysis utilizing RF regression in order to rank the importance of the aforementioned features in estimating FOC.

Table III: Feature ranking using RF

Ranking	Feature	Importance
1	STW	0.94
2	WS	0.13
3	DRAFT	0.011
4	VSL_H	0.005
5	COMBH	0.0058
6	SWH	0.0054
7	CS	0.004
8	$WAVE_H$	0.0039
9	SWP	0.0036
10	COMBD	0.0032
11	SWD	0.0028

Besides selecting the most important (i.e. informative) features, we also aim to avoid selecting highly correlated ones. For this purpose, we utilize the Spearman's Rank Correlation (SRC) coefficient, which assesses the strength and direction of the monotonic relationship between two ranked variables $R(X_i), R(Y_i)$ using covariance and standard deviation σ, and is calculated as follows:

$$\rho_{R(X),R(Y)} = {}^{cov(R(X),R(Y))}/_{\sigma_{R(X)}\sigma_{R(Y)}} \quad (3)$$

Assembling the ranking of features depicted in Table III and the correlation depicted in Figure 2 using Algorithm 1 we conclude with a subset of the initial feature set that combines feature importance and independence.

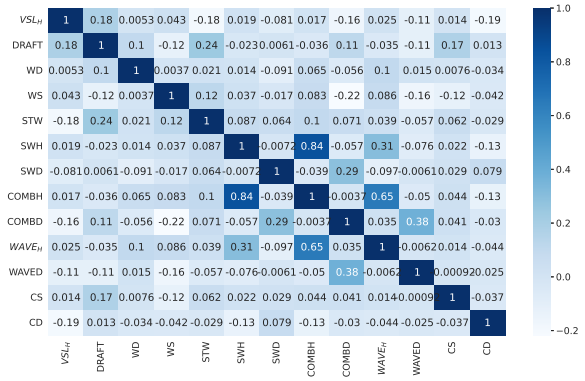


Figure 2: Spearman correlation heatmap

Algorithm 1 Feature selection based on RF regression importance and Spearman Correlation.

Require: `featureSet` $\mathcal{F} \leftarrow$ top 10 from RF
Require: `featureSet` $\mathcal{F}_r \leftarrow$ rest of features from RF
Require: `correlations` $\mathcal{C}orr \leftarrow$ from SRC
Require: `importances` $\mathcal{I}mp \leftarrow$ from RF

for each $f_i \in \mathcal{F}$ **do**
  set $\bar{\mathcal{F}} = \mathcal{F} \setminus \{f_i\}$
  for each $f_k \in \bar{\mathcal{F}}$ **do**
    if $\mathcal{C}orr(f_i, f_k) > 0.5$ **then**
      if $Imp[f_i] < Imp[f_k]$ **then**
        delete f_i From $\mathcal{F}$
        set $f_{temp} = f_k$
      else
        delete f_k From $\mathcal{F}$
        set $f_{temp} = f_i$
    for each $f_r \in \mathcal{F}_r$ **do**
      if $\mathcal{C}orr(f_{temp}, f_r) < 0.5$ **then**
        add f_r to $\mathcal{F}$ and **break**
    if $f_{temp} = f_k$ **then**
      break
Return $\mathcal{F}$

D. Data cleaning

Raw data, collected from the sensors of the vessel, are in time-series (minutely) form and tend to be "noisy" (high variance, high standard deviation from the mean) and in some cases even erroneous. 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 predictive models.

Data filtering was implemented in two stages. First, assuming that the dataset follows a normal-like distribution, we keep the data points that lie within the 99% confidence interval around the mean. Then we apply an appropriately designed Decision Tree based algorithm in order to further cancel the noise in FOC target distribution caused by the flowmeter sensor on the vessel. Then, we proceed to transform our dataset into 15-min rolling window averages in order to further smooth out any spikes and outliers that occur in the feature set from sensor installments. Note that the use of rolling window averages is consistent with the use of the FOC prediction model within a WR algorithm, in which decisions are based upon average values of FOC and not momentary consumption.

The raw data of the vessel's speed and corresponding FOC collected from the sensors, versus the mean values per speed range ($\pm 0.25V$) and the 15 min rolling window averages are depicted in Figure 4. Red circles are indicative of the number of observations found for a particular range of speed.

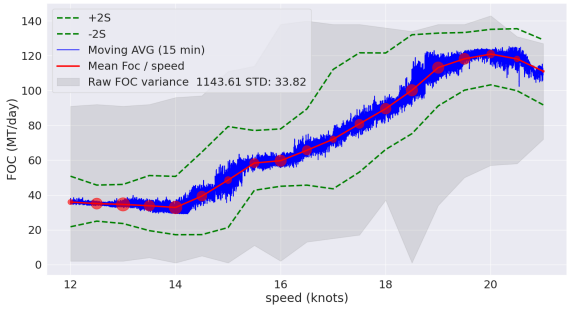


Figure 3: Raw data values VS Mean values VS Rolling window average values

Decision Trees (DTs) for data cleaning. In the first step of the data cleaning algorithm, we construct a Decision Tree (DT) with three splits, corresponding to the three most important features, according to Section III-C, namely vessel speed (STW), draft and wind speed. For the first split of the DT root node we employ the vessel's speed distribution. The child nodes are further split using Draft, and the resulting leaf nodes contain:

- The average value of FOC for a particular speed, draft and wind speed combination
- The standard deviation of the FOC for this combination
- Number of observations found for this particular weather/vessel state combination

The structure of the tree outlines the "acceptable" FOC bounds for all the possible vessel/weather state combinations extracted from a representative dataset recorded for approximately one year. After the tree is constructed, it is utilized as a support decision tool in order to classify FOC values as outliers (i.e. lying outside of the 99% confidence interval) of the FOC values kept in the leaf nodes of the tree. With this process we eliminate or replace FOC values from the initial training dataset that will most likely compromise the accuracy and the generalization capabilities of our FOC predictive scheme in the long run.

A visual abstraction of the decision tree for a sample path corresponding to ranges of: **STW** [11-14 knots] $\rightarrow$ **DRAFT** [6-11 m] $\rightarrow$ **WINDSPEED** [0 - 10 m/s] is depicted below alongside with the calculation of the average and standard deviation of FOC and the number of instances (n) found for this particular path.

IV. Model Implementation

A. FOC estimation model

The dynamic estimation of FOC based on vessel state and environmental conditions can be examined as a multivariate

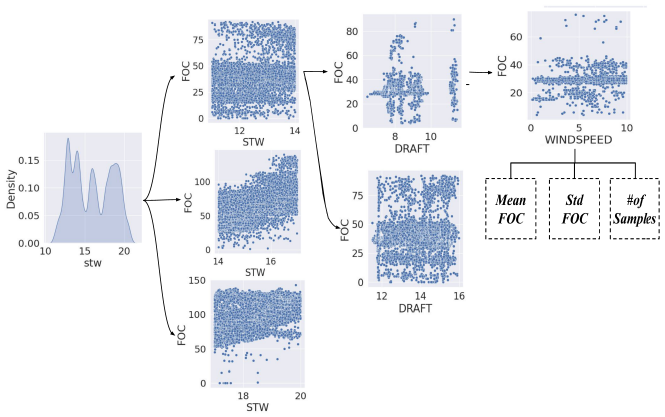


Figure 4: Decision Tree example for data cleaning

time-series prediction problem that takes into account the actual values as well as their recent history, and captures the information hidden in the values' evolution over time. Based on the superiority of Long Short-Term Memory Neural Network (LSTM) models over traditional time-series prediction methods (e.g., ARIMA) [15], LTSMs are chosen as the basis of our solution. The initial feature set, collected by AIS and sensor installments comprises the vessel speed, draft and heading and some basic weather features such as wind speed and direction. In order to take maximum advantage of this limited feature set, we employ a novel LSTM architecture, using a pre-training step that extracts information from the original features, using spline-based regression [16]. In what follows, we describe how LSTM is used for FOC estimation and detail the proposed $SplineLSTM$ model and its novel aspects.

1) The basic LSTM for FOC estimation: LSTM is a variation of traditional Recurrent Neural Network (RNN) architecture [17], which has been extensively used for time-series prediction tasks [18]. Unlike standard feed forward neural networks, LSTM also contains feedback connections and can process single data points (e.g. images) as well as entire sequences of data (e.g., speech, video or object trajectories). Compared to RNNs, Hidden Markov Models and other sequence learning methods, LSTMs are not so sensitive to the length of gaps between important events in a time series, which makes them more preferable in numerous applications. To this end, we adopt an LSTM architecture for the prediction of FOC values.

The input of the LSTM network at timestep t_u comprises N time-series, one for each feature of interest (speed through water, wind speed, wind angle etc) and in order to use the recent history of values in each feature, we employ a fixed-length time-window (time-lag of length m). As a consequence, the window contains the values for each time step $t_i \in [t_{k-m}, t_u]$ for the weather and vessel state features that are used for the estimation of FOC at time t_u, resulting in N time-series, of length $m+1$, of the form $[F_{N(u-m)}, ..., F_{N(u-1)}, F_{N(u)}]$, for each feature F_N. Given a sequence of consecutive time-steps, and a multivariate feature set, we get the following correspondence between the input and the output of the LSTM:

$$\begin{bmatrix} [F_{1(u-m)} \dots F_{j(u-m)} \dots F_{N(u-m)}] \\ \vdots \\ [F_{1(i)} \dots F_{j(i)} \dots F_{N(i)}] \\ \vdots \\ [F_{1(u)} \dots F_{j(u)} \dots F_{N(u)}] \end{bmatrix} \longrightarrow \begin{bmatrix} FOC_{u-m} \\ \vdots \\ FOC_i \\ \vdots \\ FOC_u \end{bmatrix}, \quad (4)$$

where N is the number of monitored features (and respectively of the time-series fed to the LSTM), m is the window length, and $F_{j(i)}$ is the value of feature $j \in [1, N]$ at timestamp t_i. FOC_i is the FOC value that we want to predict.

2) SplineLSTM*:* In a previous work [1], we demonstrated the approximation capabilities of spline-based regression models [19] and their ability to adapt to the linear and non-linear patterns that exist between dependent and independent variables, as those that describe the underlying function that approximates FOC. A spline regression of degree d partitions the input space in sub-domains separated by k knots. Each domain is approximated by different polynomial of degree up to d. Splines of order d have continuous $d-1$ derivatives, a property that balances the trade off between goodness-of-fit and smoothness of the spline interpolant, and results in a predictive scheme with good generalization capabilities.

An example of spline regression and the polynomials that are constructed in training time given a multi-variate feature set: $x_1, \dots, x_N$ and a target variable y is described as follows:

$$y = f(x) = \begin{cases} H_{1i}(x_{1i}, ..., x_{1N_i})b_{1i}, & x_{1i} \in [t_i, t_i + \Delta t] \\ \vdots & \\ H_{ki}(x_{ki} ... x_{kN_i})b_{ki}, & x_{ki} \in [t_i + (k-1)\Delta t, t_i + k\Delta t], \\ 0, & otherwise \end{cases} \quad (5)$$

where t_i is the corresponding time-step of observation x_i, k is the number of knots of the pre-trained SplineModel$_i$, $H(x)$ are piece-wise continuous hinge functions of order $d \geq 1$ defined on subsequent time intervals, and b_{in} are the regression coefficients of the pre-trained spline models, where $i \in [1, k]$.

The proposed spline-based LSTM uses the knowledge gained in the pre-spline training step for providing an extended input vector of size $k+N$ in each case, where k is the number of knots of the respective spline function. The proposed Spline network evaluates each feature x_i on the corresponding hinge function H_i generated from Spline regression creating a k dimensional vector that quantifies the impact of each feature in FOC estimation. In the Spline network we evaluate each feature x_i on the corresponding hinge function H_i generated from Spline regression creating a k dimensional vector that quantifies the impact of each feature in estimating FOC. The basic extension compared to a conventional LSTM is that by introducing this k-dimensional spline-informed vector the network is able to take into account not only the temporal but also the spatial structure of the features. This approach guides the network to form spatial-aware embeddings that help

the model learn a different set of functions for different sub-domains of interest.

With this transformation, the LSTM *looks back* m time steps to form the hidden state units h_{t-1}. The hidden state acts as the NN memory, for it holds information on data the network has *seen* before. The input vector is constructed by moving time windows that comprise:

- $(F_1, \ldots, F_N)$ values,
- k values generated from evaluating our feature set $(F_1, \ldots, F_N)$ values at each of the k knots of the pre-trained Spline model, and
- the corresponding FOC values.

The resulting input vector at time instance t_i takes the form:

$$\begin{bmatrix} [F_{1(1)}, \ldots, F_{N(1)}, H_{1(1)}(F_1), \ldots, H_{1(k)}(F_{1'})] \\ \vdots \\ [F_{1(i)}, \ldots, F_{N(i)}, H_{i(1)}(F_i), \ldots, H_{i(k)}(F_{i'})] \\ \vdots \\ [F_{1(m)}, \ldots, F_{N(m)}, H_{m(1)}(F_m), \ldots, H_{m(k)}(F_{m'})] \end{bmatrix} \rightarrow FOC_i \quad (6)$$

where m is the number of the previous time steps used to form the initial $2D$ vector of velocity and its mean value, m is the step used to set-up the time window vectors for the hidden LSTM units and H_{ik} is the i-th Hinge function of the pre-trained spline regression model.

V. Experimental Results

In this section we will first demonstrate results corresponding to different voyages of a real container vessel utilizing the proposed FOC estimation model. Furthermore, we briefly introduce the WR algorithm that has been utilized to validate our approach by utilizing the SplineLSTM neural network as a cost function to find the most rewarding alternative waypoints in terms of FOC. Finally, we showcase the results and the potential fuel savings during a voyage by utilising the WR algorithm as a decision support tool in order to propose an alternate route.

A. Dataset

All the experiments were conducted with real data, from a dataset of an existing container ship vessel with a carrying capacity of 3000 TEUs [1]. The values collected correspond to a vast majority of different round-trip voyages at different periods and geographical locations. As a whole, the dataset extracted for the purpose of this work, covers a time span of one year (December 2019 - December 2020) with approximately $4 * 10^5$ data points.

In order to examine the statistical significance of our results, we created 10 statistically independent subsets extracted from different time periods of approximately $5 * 10^3$ observations each that cover 84 hours or 3.5 days of the vessel's trip.

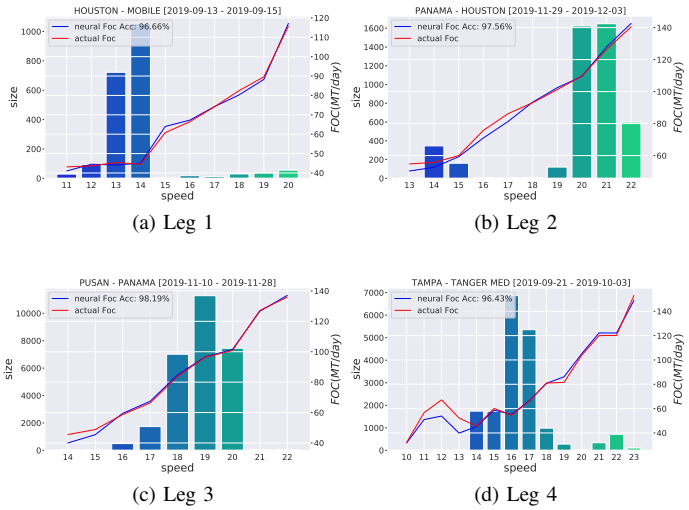
From these datasets, 80% was used for training and the rest for testing. Statistical independence was preserved between different datasets with the use of the Kolmogorov-Smirnov test (KS-test). This is a two-sided test for the null hypothesis that 2 independent samples are drawn from the same continuous distribution. The dataset used in the context of this work is available, in sanitized form, upon request to the first of authors. It contains the values for the features described in Table I, and their corresponding timestamp.

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

B. SplineLSTM *Performance in different voyages*

The next step is to evaluate the approximation capabilities of the proposed LSTM-based FOC predictive model. This is performed for four different voyages extracted from the initial test set. The voyages correspond to different locations, time periods and weather conditions for the same container ship.

To demonstrate the results, we depict, in Figure 5 and Table IV, the deviation between the actual and the predicted FOC measured in Metric Tones for one day (MT/day), per speed(V) range ($\pm 0.5V$). Bar size indicates the number of observations found for a particular speed range.



(a) Leg 1 (b) Leg 2

(c) Leg 3 (d) Leg 4

Figure 5: *SplineLSTM* performance in 4 different voyages of the same container ship

Table IV: Computational performance of the FOC-model (*SplineLSTM*)

	Act. FOC	**Pred. FOC**	**Abs. Diff**	**Perc. Diff (%)**
HOUSTON - MOBILE	71.84	72.68	0.84	1.16
PANAMA - HOUSTON	360.24	362.04	1.8	0.5
PUSAN - PANAMA	1725.14	1714.98	10.16	0.59
TAMPA - TANGER MED	774.53	771.45	3.08	0.4
Total	**2921.57**	**2931.94**	**10.56**	**0.36**

Computational performance superiority of the *SplineLSTM* model, as presented above, allows us to utilize it in the context of a weather routing optimization algorithm.

C. *Coupling* SplineLSTM *with a WR algorithm*

To validate further our approach in the context of a real world application, *SplineLSTM* has been coupled with a WR algorithm to support vessel routing decisions towards the reduction of FOC. The WR algorithm that has been utilized is based on the isochrone principle [20]. It builds upon a predetermined basic route; this route can be the original route planned by the vessel's master or provided by a basic routing algorithm. In the context of this work an initial route was employed on the basis of shortest path principles. The original (initial) route is then broken into segments, with respect to a given time step (indicating the master's routing decision horizon, e.g., every 6 hours), and a graph is built around it that enables course and speed deviations, while "following" the direction of the vessel's original course. To this end, for each node of the original route, a set of nodes is added in a "parallel" fashion on both sides of the route (i.e., parallel to the direction of the original route). Edges are added between all nodes of subsequent sets. Note that nodes that are identified to be on land as well as edges that go above land segments are naturally excluded from the graph.

Once the graph is created (Figure 6), *SplineLSTM* is used to obtain the FOC of each edge of the graph, i.e., of each corresponding sea route, given the vessel's STW, draft and corresponding weather conditions along that sea route. After scoring each sea route (i.e., graph edge), a variation of Dijkstra's algorithm for the shortest path problem is utilized to obtain the route that minimizes the total route FOC (i.e., considering the calculated FOC of each edge as its corresponding "edge weight" or "distance"). Note that since the algorithm is isochrone, the produced route also satisfies any constraints concerning the time of arrival (if any). Note also that the decision variables for the WR algorithm are only the STW and the vessel's direction, since these are the aspects that the vessel's master can control. Obviously, any change in the vessel's speed affects directly FOC (since STW is a basic feature of the corresponding model). However, changes in speed and direction also affect FOC indirectly, since they alter the spatio-temporal state of the vessel and hence the corresponding weather conditions.

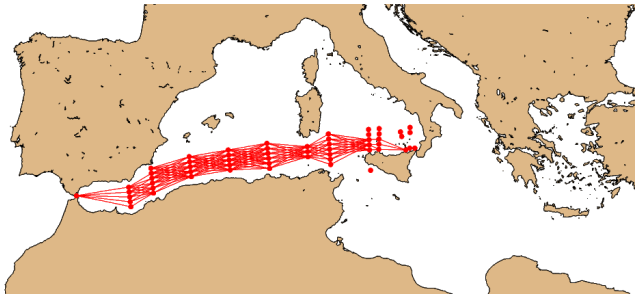


Figure 6: Graph construction comprised of alternative waypoints(red circles) for an example route

VI. Weather Routing Optimization Experimental Results

In this section we demonstrate results of the WR optimization algorithm presented in section V-C. We compare the total FOC of an initial transatlantic voyage conducted by the vessel's master, with the suggested optimized route produced from the WR algorithm by utilizing *SplineLSTM* model. Furthermore we calculate the total distance travelled, the estimated time of arrival, the average speed and the emissions emitted for the two alternative routes by incorporating Table II, and we exhibit the results in Table V. Consecutively we demonstrate the weather (wind speed (m/s)) of the initial and the optimized route per hour, in Figure 8.

Figure 7: Initial (blue) and Optimized (red) route for one leg: TAMPA (FLORIDA U.S) - TANGER MED (MOROCCO)

Voyage	Date	Latitude	Longitude
Departure	2019-09-21	27.7° N	82.5° W
Arrival	2019-10-03	35.8° N	6° W
BASIC COMPARISON	**Actual route estimation**	**Optimized route estimation**	
Distance (nm)	4787.4	4369.36	
Time (hours)	289.97	264.63	
Avg Speed (kt)	16.52	16.51	
Total FOC (MT)	774.53	759.97	
CO_2 (MT)	2411.88	2366.54	

Table V: Estimation based on Weather Service (NOAA)[2]data

VII. Conclusions and Next Steps

This work introduced a framework for real time data collection and processing, related to vessels operations, in order to employ a range of multi-purpose data driven schemes. This dynamic environment aspires to support in decision making different professions related to ship operations and maintenance, like brokers, operators, marine engineers, the crew of the vessel, etc.

In this paper the pipeline was adapted accordingly, firstly to extract an ideal feature set and then to employ a robust Deep Learning model (*SplineLSTM*) for FOC estimation purposes.

In continuance we showcased the approximation capabilities of the *SplineLSTM* model and we demonstrated the environmental footprint of the joint *SplineLSTM*-WR optimization algorithm by proposing an alternative route with lower FOC, and therefore reduced CO_2 emissions for a particular voyage.

[2]weather features where acquired from National Oceanic and Atmospheric Administration (NOAA)

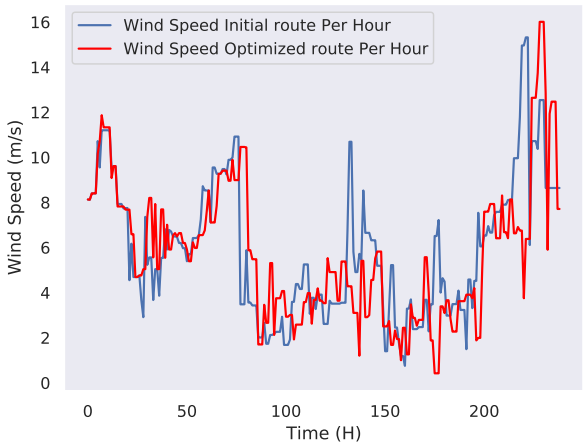


Figure 8: Weather comparison (wind speed) for the initial and optimized route of Figure 7

One of the main directions to expand this work, is to utilize the proposed framework to collect and eventually store a broader category of variables comprised of vessel's particulars (propeller diameter, hull geometry, main engine type, fuel type etc). This will aid us to build a physics informed library for different types of vessels. This dataset can then be employed for incremental and transfer learning purposes between different fleets of vessels, methods that aspire to tackle one of the main obstacles maritime industries are facing nowadays, that is the lack of historical data for many vessels.

Acknowledgment

This Publication was supported partially by the program of Industrial Scholarships of Stavros Niarchos Foundation, and partially by the SmartSea project, an European Union's Horizon 2020 research and Innovation programme under the Marie Skłodowska-Curie Grant Agreement No 612198. Access to industrial data has been provided by Danaos Shipping Co.

References

[1] D. Kaklis, G. Giannakopoulos, I. Varlamis, C. D. Spyropoulos, and T. J. Varelas, "A data mining approach for predicting main-engine rotational speed from vessel-data measurements," in *Proceedings of the 23rd International Database Applications & Engineering Symposium*, 2019, pp. 1–10.

[2] A. Coraddu, L. Oneto, F. Baldi, and D. Anguita, "Vessels fuel consumption forecast and trim optimisation: a data analytics perspective," *Ocean Engineering*, vol. 130, pp. 351–370, 2017.

[3] M.-I. Roh and K.-Y. Lee, *Computational ship design*. Springer, 2018.

[4] B. Kim and T.-W. Kim, "Weather routing for offshore transportation using genetic algorithm," *Applied Ocean Research*, vol. 63, pp. 262–275, 2017.

[5] C. Gkerekos and I. Lazakis, "A novel, data-driven heuristic framework for vessel weather routing," *Ocean Engineering*, vol. 197, p. 106887, 2020.

[6] C. Walsh and A. Bows, "Size matters: exploring the importance of vessel characteristics to inform estimates of shipping emissions," *Applied Energy*, vol. 98, pp. 128–137, 2012.

[7] M. M. Golias, G. K. Saharidis, M. Boile, S. Theofanis, and M. G. Ierapetritou, "The berth allocation problem: Optimizing vessel arrival time," *Maritime Economics & Logistics*, vol. 11, no. 4, pp. 358–377, 2009.

[8] T. Varelas, S. Archontaki, J. Dimotikalis, O. Turan, I. Lazakis, and O. Varelas, "Optimizing ship routing to maximize fleet revenue at danaos," *Interfaces*, vol. 43, no. 1, pp. 37–47, 2013.

[9] Y. W. Shin, M. Abebe, Y. Noh, S. Lee, I. Lee, D. Kim, J. Bae, and K. C. Kim, "Near-optimal weather routing by using improved a* algorithm," *Applied Sciences*, vol. 10, no. 17, p. 6010, 2020.

[10] Z. Zhao, K. Ji, X. Xing, H. Zou, and S. Zhou, "A novel method for joint optimization of the sailing route and speed considering multiple environmental factors for more energy efficient shipping," *Ocean Engineering*, vol. 216, no. 2, p. 295, 2020.

[11] M. Jeon, Y. Noh, Y. Shin, O.-K. Lim, I. Lee, and D. Cho, "Prediction of ship fuel consumption by using an artificial neural network," *Journal of Mechanical Science and Technology*, vol. 32, no. 12, pp. 5785–5796, 2018.

[12] C. Gkerekos, I. Lazakis, and G. Theotokatos, "Machine learning models for predicting ship main engine fuel oil consumption: A comparative study," *Ocean Engineering*, vol. 188, p. 106282, 2019.

[13] Z. A. Papandreou Christos, "Predicting vlcc fuel consumption with machine learning using operationally available sensor data," *Ocean Engineering*, vol. 197, p. 106887, 2020.

[14] Z. Yongjie, Y. Zuo, and T. Li, "Predicting ship fuel consumption based on lstm neural network," *2020 7th ICCSS Conference*, vol. 32, no. 12, pp. 310–313, 2020.

[15] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE ICMLA*. IEEE, 2018, pp. 1394–1401.

[16] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, pp. 1–67, 1991.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.

[19] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[20] G. Hanssen and R. James, "Optimum ship routing," *Journal of Navigation*, vol. 13, pp. 253–272, 1960.