New Insights into the Propulsion Power Prediction of Cruise Ships

Fred Gonsalves, Bastien Pasdeloup, Romain Billot, Patrick Meyer, Arnaud Jacques, Matthieu Lorang

New Insights into the Propulsion Power Prediction of Cruise Ships

Fred Gonsalves*†, Bastien Pasdeloup*, Romain Billot*, Patrick Meyer*, Arnaud Jacques† and Matthieu Lorang†
* IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest F-29238, France
Email: fred-michael.gonsalves@imt-atlantique.fr and firstName.lastName@imt-atlantique.fr
† Chantiers de l'Atlantique
Email: firstName.lastName@chantiers-atlantique.com

***Abstract*—Ship propulsion is the largest consumer of energy – and by extension fuel – on cruise ships. Improving its efficiency is thus an important aspect of energy management, both for environmental and economic reasons. Various approaches have been detailed in the literature for improving propulsion efficiency, ranging from optimal voyage planning to prediction of propulsion power or fuel consumption using Machine Learning algorithms, trained on high frequency sensor data. On this latter topic, the approaches typically involve a series of data transformations and time-aggregations (windowing), followed by shuffling and separation of data points into train and validation sets. However, this approach leads to very similar data in the train and validation sets, preventing trained models to generalize well on future ship voyages. In this article we highlight methodological issues and give insights on how to tackle them to train models that focus on optimizing generalizability, especially predictive accuracy on unseen future test sets. We present a temporal approach to splitting data into train, validation and test sets. We perform our analysis using simple multilayer perceptron architectures, of distinct dimensions. Our study concludes that smaller/simpler models, trained on temporal-split data have a lower error when predicting on unseen future test data, compared to larger models and usage of shuffle-split datasets, while also providing better confidence in model accuracy, due to reduced discrepancy between obtained validation and test errors.**

***Index Terms*—propulsion power prediction, methodology, neural networks.**

I. Problem Statement

A. Context and Challenges

Global shipping accounts for roughly 2.8% of global carbon emissions, according to the International Maritime Organization 2014 Green House Gas study [1]. The current objective of the organization is to reduce shipping carbon intensity by 40% by 2030 and 50% total (70% intensity) by 2050 compared to the 2008 baseline [1]. It specifies evaluation indexes, as well as milestones to be achieved, based on the type and size of the ship. Emissions of other pollutant chemicals such as sulfur are also increasingly being tightened by restrictions internationally and in protected zones [2].

Shipping companies and shipyards alike are thus under pressure to reduce fuel and energy consumption and, consequently, carbon emissions to comply with these relatively recent objectives and evolving regulations. In this context, cruise lines play a crucial role in reducing their greenhouse gas and chemical emissions owing to size and number of passengers. Cruise lines and shipyards work in tandem to innovate fuel-saving solutions involving all phases from design to operation of the ship. While a cruise ship can be compared to a small city in the number and variety of energy consumers on board (e.g., air conditioning, hotel, water production, lighting, entertainment infrastructure), the major fuel consumer of a cruise ship is the propulsion system.

The propulsion of a cruise ship accounts for roughly 40-60% of the ship's total energy consumption [3]. The performance of the ship in terms of its energy efficiency is thus highly dependent on this system, and efforts are being made in various directions to improve it. Energy efficiency improvement strategies can roughly be classified as either *design* or *operational*. Design improvements may include hull form, propeller design, energy conversion efficiency, use of greener fuels and alternative energy sources (e.g., liquefied natural gas, methanol), or supplementary equipment (e.g., air lubrication system). From an operational perspective, measures such as speed and trim optimization, best route prediction, or regular hull maintenance can lead to important energy savings. With respect to the latter point, hull fouling – i.e., accumulation of marine growth on the hull – could indeed lead to an increasing performance deterioration and hence an increase in energy consumption [4].

Analysis and evaluation of energy-saving measures has been boosted by the improvement of data collection, storage and processing technologies. The literature today contains various examples of monitoring tools capable of collecting high frequency data from various sensors and analyses are being carried out over several months of data. Among these tools, Machine Learning (ML) has gained a lot of interest, especially models based on Artificial Neural Networks (ANNs). However, it appears that most works from the field of propulsion power prediction suffer from methodological issues, which is the main message of this article.

B. Related Work

In recent years, various studies have analyzed the use of ML algorithms applied to ship propulsion. The problem is typically stated as a prediction problem with the target variables being propulsion power [5], [6], [7], fuel consumption [8], [9], or speed [10], [11] and the input variables being operational parameters such as ship trim, draft, speed, and external conditions such as wind and sea state. The datasets used are either

noon reports (data sheets prepared on a daily basis by a ship's chief engineer, consisting of average values of the above-mentioned parameters, as well as the total fuel consumption and other information) or ship sensor data, sampled at varying frequencies, weather forecast or hindcast data. In the field of propulsion power prediction, the first ML-based approach – based on sensor data – was introduced in 2009 [5]. However, the use of such data has only seen an increase starting around 2017, and the number of papers on the topic yet remains limited.

Although data can be viewed as signals or time-series, a common simplification of the problem – assumed by the majority of articles to date – consists in considering data as a collection of independent observations. While this certainly reduces the accuracy of predictions, it allows to evaluate first the predictive powers of features without considering their dynamics. In this article, we choose to make the same simplification, as we want to focus on methodological issues in the literature. Considering time series, though, is a clear line for future work.

The standard approach taken in the literature for preprocessing data before training and testing ML models thus consists of the following steps:

1) applying certain filters [12];
2) extracting features;
3) performing appropriate aggregations (e.g., mean, variance, derivative) of the input and target variables.

The data points are then split into train, validation, and/or test sets. Although, performing this split is approached in various ways in the literature. The most common one is the *shuffle-split* approach, in which the processed data points are randomly shuffled and then allocated to these subsets.

Pedersen *et al.* [5] and Petersen *et al.* [6] are the first studies focusing on ANNs and sensor data, collected at a high frequency (up to 1Hz in [5]). Data is then aggregated using a fixed window length. The architecture implemented in both cases is a MultiLayer Perceptron (MLP), with hyperparameters selected using a parameter sweep and k-Fold Cross-Validation (k-Fold CV). Petersen *et al.* [6] propose two methods for shuffling data: 1) shuffling individual 10-minute windows; 2) shuffling trips. The authors report that shuffling trips leads to a slightly higher prediction error, but that this method avoids what they refer to as *cross-talk*, whereby the data in the train and test sets are likely to be very similar. This is very reasonable, since the dynamics of large moving ships are long-term and consecutive ten-minute aggregated windows of data are not likely to be very different from each other.

Drawing on the conclusions of [5], Du *et al.* [13] use a MLP architecture on noon report data. The goal of their approach is to quantify the impact of sailing speed, trim, displacement, weather and sea conditions on a ship's fuel consumption rate and then to optimize speed and trim for each segment of a voyage in order to minimize the total fuel consumption.

Kim *et al.* [8] study the use of MLP and multiple linear regression models to predict the fuel oil consumption of a container ship using 6 months of sensor data. Training of the MLP model is performed using train/validation sets generated by a shuffle-split approach. The authors then test the model on an independent voyage not used at all during training. Similar settings are also used in the work of Farag et. al [14], in which they train an ANN on a first voyage and report test errors on a distinct second one, for an oil tanker.

Gkerekos *et al.* [15] evaluate the use of various ML algorithms to predict fuel oil consumption using both noon report data and sensor data collected using an automatic data logging and monitoring system, sampled every hour. The algorithms include, among others, linear regression, tree- and forest-based models, k-Nearest Neighbors (k-NN) and MLPs. Separate models are trained for each type of dataset. The noon report dataset contains two and a half years of data, while the other contains only three months of data.Each dataset is then split into train and test subsets (80%-20%) and k-Fold CV is performed on the train set.

Uyanik *et al.* [9] also perform a comparison of various ML models for the prediction of fuel oil consumption. Here, data comes from noon reports and the target variable is the total fuel oil consumption over the voyage. The methodology used is similar to the shuffle-split approach. Though, no aggregations needs to be performed as the information is already aggregated in the noon reports. Grid-search and k-Fold CV are used to find optimal hyperparameters for the different ML models.

Beşikçi *et al.* [16] implement a MLP architecture to predict fuel oil consumption from noon report and weather data. The data points are shuffled and split into train and validation sets (70%-30%). Hyperparameters for the MLP were selected using trial-and-error.

Theodoropoulos *et al.* [7] evaluate the use of two ANNs for the prediction of ship propulsion power using sensor data. First, a MLP architecture similar to the previous literature is used. Data is preprocessed using various filtering and outlier removal techniques. Data is then smoothed using a simple moving average over 5-minute windows. Training is performed following the shuffle-split approach and using k-Fold CV. The final model is then used in a simulated operational scenario where the first 80% of the data is used for training and the final 20% data for testing. The article evaluates a second neural network model for propulsion prediction using Long Short-Term Memory (LSTM) architecture. However, as mentioned earlier in the section, we will focus on models not incorporating the temporal aspect of the data.

Finally, Laurie *et al.* [4] tackle the subject of propulsion power prediction using ML models. Their study evaluates various ML models, trained with sensor data (10 second sampling rate) and weather data. The authors provide a model to analyze the effect of hull fouling on the propulsion performance of the ship. They use a shuffle-split approach after aggregating over 10 minute intervals, followed by hyperparameter selection using k-Fold CV. Laurie *et al.* point out that a k-NN algorithm achieves the second best performance with a Mean Absolute Percentage Error (MAPE) only 0.07% lower than the best model – a random forest.

C. Knowledge gaps and contributions of the paper

The state of the art raises several issues regarding a potential implementation of prediction models in operational conditions:

1) **The cross-talk problem** [6]: Separating high frequency, sensor-acquired data into train and validation subsets *after* shuffling would always lead to very close analogs in the train set for every validation data point (see Figure 1). An issue thus arises when using this shuffle-split approach for validating the predictive accuracy of the model: although the validation error reported for a model may be very low, it is possible that the model is strongly overfitting to training data, leading to high accuracy errors in testing conditions.

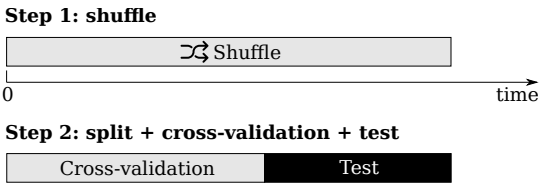

Fig. 1. The cross-talk problem

An indication of the existence of this cross-talk issue is the performance of the k-NN algorithm in evaluation papers [15], [4]. In particular, Laurie *et al.* point out that for any data point in the validation/test sets, there exist very close analogs in the train set, when data is sampled at a high frequency [4]. Figure 2 illustrates this phenomenon.

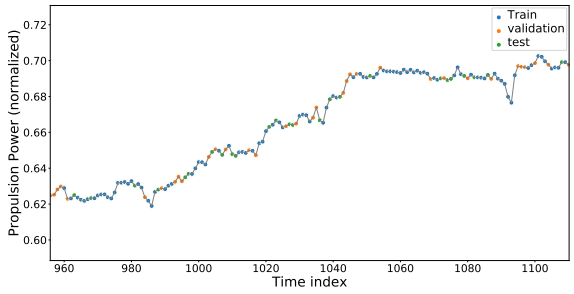

Fig. 2. Shuffled data

2) **Overlapping windows**: The problem of cross-talk is further exacerbated when overlapping windows are used for aggregation, such as in the case of simple moving averages [7]. Such an operation, used in conjunction with a shuffle-split approach, causes train and validation datasets to share information, leading to trained models not being able to learn generalizable characteristics of the data. Thus, validation accuracy reported may be misleading if the goal is to implement models for predicting ship behavior for future voyages.
3) **Parsimony**: the literature review exhibits already a wide range of ML models, from simple (e.g., multiple linear regression) to complex (e.g., recurrent neural networks) approaches. Laurie *et al.* [4] suggest that parsimonious models such as k-NN can outperform most refined techniques for the prediction task. Therefore, one gap is related to a rigorous methodology to compare models' complexity, for instance by comparing small vs. large architectures within the same models family. As an example, a small MLP performing at least as well as a large model on future unseen data would indicate overfitting in the selection of the larger one. Moreover, overfitting could be detected when a model performs much worse on future, unseen test data than could be reasonably expected. Concretely, the test error is much higher than the validation error.

The goal of this paper is *NOT* to propose a new comparison of ML methods for propulsion power prediction. The purpose is rather to come up with a simple methodological contribution that addresses the above-mentioned knowledge gaps. The research questions we would like to tackle are three-fold:

Q1) Given a propulsion power prediction task, what is the most appropriate splitting approach (train/validation/test) to obtain a model leading to similar validation and test errors (i.e., providing more confidence for implementation in operational conditions)?
Q2) Given a fixed size of a model, which splitting approach gives the minimum test error (i.e., which approach generalizes better)?
Q3) Given a specific splitting approach, does a small model perform as well on a future test as a large model (i.e., principle of parsimony)?

With respect to these research questions, the main contributions of this paper can be summarized as follows:

- **Methodological framework**: We propose to compare two methodologies to train and validate a machine learning model for prediction of propulsion power using 27 months of high-frequency sensor data. The first of these is the *shuffle-split* approach, whereby data for *both* train and validation datasets comes from exactly the same timeframe (e.g., 1 year). In the second approach, data is split temporally (we refer to this as the *temporal-split* approach); the first part of the dataset is used for training, the second part for validation.
- **Models dimension analysis:** Based on a MLP architecture, we also analyze the dimensions of the models obtained using the two approaches, and propose a *cross-comparison* of small vs. large MLP configurations in both splitting strategies (shuffle vs. temporal split).
- **Realistic evaluation**: The best models obtained from the two approaches are then tested on an independent future test set, unseen by either model during the training process. This allows us to clearly see the gaps between validation and test errors (hence assessing overfitting) and also highlight the degradation of the performance when the prediction horizon increases.

The paper is organized as follows. In Section II, we present the global methodological framework and the splitting strategies. Section III is related to the data presentation, including preprocessing, feature engineering and windowing. Results are presented in section IV. Section V concludes the paper with a discussion.

II. Systematic methodology

To answer the three research questions **Q1**, **Q2**, **Q3** presented in Section I-C, we propose a global methodological framework that enables a comparison of the two splitting strategies for training and validation, as well as a cross-comparison of small vs. large MLP architectures. The process starts with the selection of a machine learning framework.

A. Choice of the ML framework

A MLP architecture was selected as the ML framework to be used in this study. This choice was made for several reasons:

1) This approach is majoritary in the literature on propulsion power/fuel consumption prediction;
2) The dimensions of two MLP models can easily be compared in terms of number of weights to train, related to number of layers and layer size;
3) Existing work tends to use large MLPs (*e.g.*, [7]). We want to explore the use of smaller networks and their impact on prediction.

MLP can have various architectures, defined by hyperparameters such as the number of layers and nodes (neurons), or choices of activation functions. While there is a contemporary dominance of neural networks and deep learning in the ML community, the goal of this paper is not to come up with the best architecture for propulsion power estimation – e.g., including convolutional layers and temporality (e.g., LSTMs). Hence, we proceed to use simple MLP architectures and rather maintain focus on the learning process. For simplicity, we use an architecture where the number of nodes per layer is constant, and find optimal hyperparameters using a grid search procedure.

B. Global methodology

Given an appropriate ML framework, Figure 3 presents the flowchart diagram of the different steps performed during experimentation.

As shown in Figure 3, the systematic methodology is composed of the following blocks:

(A) **Data preprocessing:** consisting of feature transformations and filters presented in Section III-B;

(C, D) **Data splitting:** divide the dataset into train, validation and test sets according to the two possible approaches: shuffle-split and temporal-split, described in Section II-C;

(B, E) **Windowing and aggregation:** described in Section III-C. Depending on the selection strategy (shuffle vs. temporal), this step comes before or after the split;

(F, G) **Train model (grid search):** for both temporal- and shuffle-splits, the optimal models and their hyperparameters are found;

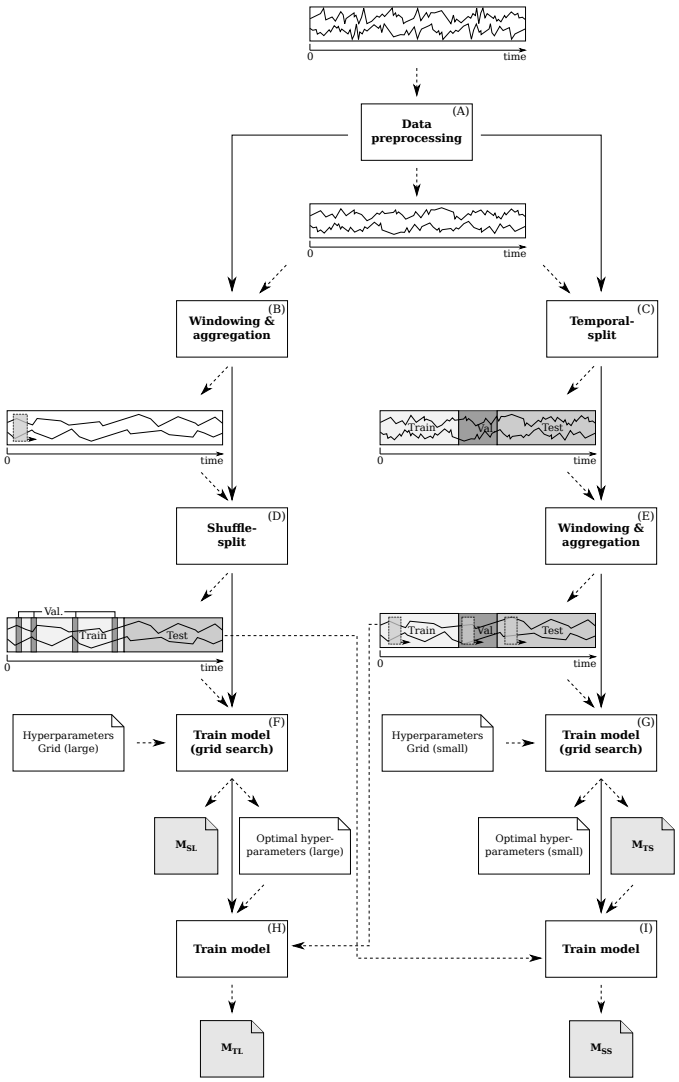

Fig. 3. Global methodological framework

(H, I) **Cross-training:** the purpose is to use the optimal hyperparameters obtained using one approach (e.g., shuffle-split) to train a model using the dataset of the other approach (e.g., temporal-split), yielding to two extra models (described in Section II-D).

All four models are evaluated on an independent future test set, unseen by the models during the training process.

C. Shuffle vs. Temporal Split

Figure 4 depicts the two strategies taken for model training and evaluation – shuffle-split and temporal-split – for preparing the train and validation sets.

1) **Shuffle-split approach:** In this first approach, all raw training data is used both for training and validation. This is performed by shuffling the data points after aggregation, as described in Figure 4;

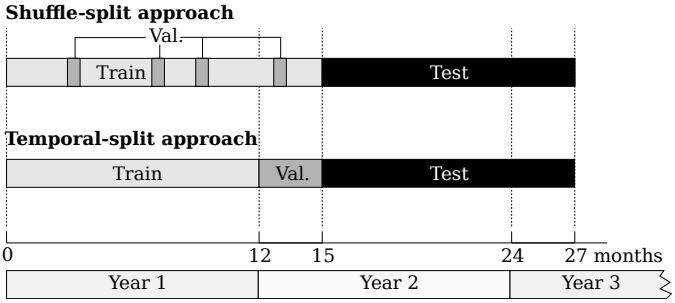

Fig. 4. Comparison of shuffle-split and temporal-split approaches for train and validation sets generation. Note that the test set remains identical.

2) **Temporal-split approach:** In the second approach, data used for training and validation are selected using a temporal split as described in Figure 4.

Train and validation set sizes are set the same for both approaches. The temporal split was applied assigning the first 12 months of data for training and the next 3 months for validation.

D. Models and evaluation

We now describe the models that we obtain using the two splitting approaches – labeled (F) and (G) in Figure 3 – as well as the models obtained obtained after cross-training – labeled (H) and (I) in Figure 3 –. In terms of the MLP learning process, we use settings described in Table I.

Parameter	Value
Batch Size	128
Optimizer	Adam [17]
Learning Rate	5e-4
Max Epochs	50
Loss	Mean Absolute Percentage Error (MAPE)

TABLE I
SHARED HYPERPARAMETERS

Number of layers and number of nodes per layer were defined experimentally through grid search to find optimal hyperparameter values. We have used two distinct grids of possible hyperparameter values, deemed *large* (described in Table II, used in step (F) in Figure 3) and *small* (described in Table III, used in step (G) in Figure 3). The use of these two grids aims at addressing research question **Q3** on the principle of parsimony. Values of the large grid have been chosen to be in the same range as those used in [7].

Parameter	Possible values
Layers	10, 15, 20
Nodes per layer	50, 100, 200, 300, 400

TABLE II
HYPERPARAMETERS GRID FOR THE LARGE MODEL

For each combination of layers and nodes, k-Fold CV was used to train 4 models ($k = 4$) to account for model initialization and batches variability. The performance of the parameters is evaluated using the average validation loss over the k models obtained per parameter.

Hyperparameter	Possible values
Layers	1, 3, 5, 10
Nodes per layer	5, 10, 15, 20, 30, 40, 50

TABLE III
HYPERPARAMETERS GRID FOR THE SMALL MODEL

Models resulting from the training procedures using grid searches are labeled M_{SL} and M_{TS} in Figure 3[1]. Comparison between these two models will give us an idea of whether or not overfitting is occurring.

Next, in order to test the effect of model size independently from the splitting approach, we interchange the optimal hyperparameters found during model building and the train/validation datasets used to obtain them (step referred to as *cross-training* in Section II-B. This produces two more models, M_{SS} and M_{TL}, as shown in Figure 3.

The evaluation metric used is the Mean Absolute Percentage Error (MAPE) [18]. The network takes as input an array of sensor readings and weather forecast data and produces as output an estimated propulsion power. The network is then trained using the MAPE as the loss function to minimize.

III. DATA PRESENTATION

In this section we discuss the data that was used for the experiments, as well as the various preprocessing steps that were taken to prepare the prediction task. For confidentiality reasons, information allowing a possible identification of the ship (e.g., absolute speed and propulsion power, geographical coordinates) is not described.

A. Data description

The dataset comes from a large cruise ship in operation. Available data consist of a few dozen variables, however, only 20 of these variables with a known effect on the overall ship resistance were selected. Two primary types of data are present in the set:

1) **Sensor data:** Data coming from sensors and systems installed onboard the ship. This data is used in the automation and control systems of the ship. Data is first stored on board and then transferred to the data provider. Sensor data selected includes speed (over ground and through water), drafts (fore and aft), acceleration (over ground and through water), number of stabilizers used, air and sea temperatures, thruster power, and electrical propulsion power.
2) **Weather forecast data:** This data is acquired from a third party meteorological data provider. All interpolations are performed by the meteorological data provider, and information regarding how these interpolations are performed is not, at this time, known. Information collected includes relative wind (longitudinal and transversal speed) and wave (period, angle and height).

[1]Models are identified as M_{xy}, where $x \in \{S, T\}$ denotes the use of the *S*huffle- or *T*emporal-split dataset, and $y \in \{S, L\}$ denotes the use of hyperparameters coming from the *S*mall or *L*arge grid for training.

The dataset collected contains 27 months of data. The sensor data is sampled every 30 seconds. Figure 5 shows a sample time series of the target variable, propulsion power.

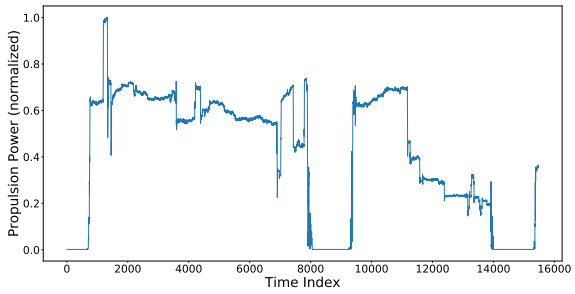

Fig. 5. Sample propulsion power time series (values are normalized for anonymity)

Accordingly, the weather data also contains data points at a 30-second sampling rate – however, as mentioned earlier, this data is interpolated spatially and temporally by the meteorological data provider as a function of the ship's geographical coordinates and time. Weather forecast data was used in this study as the model aims at being used for propulsion power forecasting purposes. Air and sea temperatures were taken from the sensor readings as they were found to be very highly correlated with the weather forecast data.

The data collected contains temporal gaps and noise issued from possibly varied sources (e.g., sensor error, extrapolation error). This requires a few basic filters and preprocessing steps to be applied before the dataset can be used.

B. Preprocessing and feature engineering

As the goal of this study is not to find the optimal preprocessing steps or features that provide the best model, but rather to focus on the implications of the methodology used, only simple operations were performed for cleaning and preparing the data:

- **Feature transformations:** Two types of feature transformations were used. For variables that are supposed to contain integer values, a rounding was applied if the values were reported with decimal points. For example the number of stabilizers used should contain values 0, 1, or 2. For variables containing angles represented in degrees (0 - 360), the sine and cosine of the angles (in radians) were extracted and used as input features;
- **Filters:** Two types of filters were used in the processing of the 30-second raw data: 1) *Operational filters:* data was filtered to remove outliers concerning the speed and propulsion power of the ship. This includes setting a minimum and maximum value for the variables concerned. Data points having thruster power greater than 0 were also eliminated from the set, as thrusters are only used during maneuvers and not at sea. These filters were applied to keep only data representative of the ship's time spent at sea and to remove periods spent at berth, maneuvers and outliers; 2) *Noise filters:* after visualizing the various data, it was found that certain sensors were giving erratic values at certain times. These data points were manually filtered out.

C. Windowing and aggregation

Windowing and aggregation has been performed in all of the studies dealing with high frequency sensor data. The reasons for these are to remove potential noise from the sensors, and also to extract features relevant to the dynamics of the system. Studies have applied various window sizes ranging from 3 to 15 minutes depending on the dataset and application. For this study, we have used a window size of 5 minutes.

Windowing produces an $M \times N$ matrix of the input variables (where M is the number of samples per window and N is the number of input variables) and an $N \times 1$ vector of propulsion power. The window is rejected if any value – either in the input matrix or in the output vector – is missing. This is done in order to not deal with data imputation and its effects.

If the window has not been dropped, the data in the window is aggregated by taking the average over the time dimension, producing an input vector of dimension $1 \times N$ and an output vector of dimension 1×1.

IV. Results

Table IV presents the final hyperparameters selected for the two splitting approaches presented in Section II-C, i.e., the output of (F) and (G) in Figure 3. These hyperparameters represent the models showing the smallest validation loss in their respective approaches.

Using the shuffle-split approach and the large grid, optimal hyperparameters found are the largest possible values from Table II (20 layers and 400 nodes per layer). The temporal-split method, trained with the small grid, produces a smaller network, with optimal values being intermediate in tested values from Table III (3 layers and 40 nodes per layer).

Model	**Layers**	**Nodes per layer**
M_{SL}	20	400
M_{TS}	3	30

TABLE IV
GRID SEARCH RESULTS: OPTIMAL HYPERPARAMETERS FOUND DURING STEPS (F) AND (G) IN FIGURE 3

According to the proposed methodology, the hyperparameters of M_{SL} and M_{TS} have been used on the datasets built by the two other splitting approaches, temporal and shuffle, respectively, leading to two extra models M_{TL} and M_{SS}. Table V reports the validation and test losses of the four models.

Model	Val. Loss	Test Loss	Difference
M_{SL}	0.75	6.33	88.3%
M_{TL}	4.45	5.41	17.7%
M_{SS}	2.75	5.52	50%
M_{TS}	4.35	5.14	15.2%

TABLE V
COMPARISON OF VALIDATION AND TEST LOSSES FOR THE FOUR MODELS

With respect to the research question **Q1**, we notice that the two models applied to a shuffle-split data set exhibit the lowest validation losses, respectively 0.75 et 2.75 for M_{SL} and M_{SS}, and yet are associated with the highest test losses (6.33 and 5.52). This result demonstrates the presence of the cross-talk problem between the train and validation sets, and the degradation of the generalization performances when the test set is appropriate. On the other hand, the temporal split allows getting more confident validation losses, compared to test losses: 4.45 vs. 5.41 for M_{TL}, 4.35 vs. 5.14 for M_{TS}.

The aim of research question **Q2** is to compare which splitting approach gives us the minimum test error, given a fixed model size. Figure 6 reports the error distributions of the four models for the 12 months of test data.

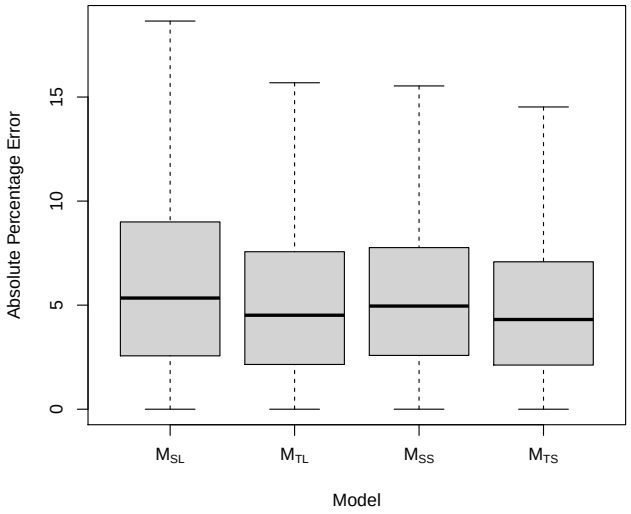

Fig. 6. Boxplot distributions of the Absolute Percentage Error for the 4 models

Comparing large size models, Figure 6 shows that M_{TL} outperforms M_{SL} with lower percentiles and less variability. The same conclusion lies for smaller architectures, as we can see the dominance of M_{TS} over M_{SS} with a comparable interquartile range. Thus, temporal-split enables a better generalization with unseen data. Regarding the research question **Q3**, whose goal was to investigate the models' dimensions, Figure 6 indicates that, for a given splitting approach, a small architecture is in nearly all cases slightly better than a large one. This can be seen by comparing M_{SL} vs. M_{SS}, as well as M_{TL} vs. M_{TS}. The significance of the differences between pair-to-pair distributions have been assessed through hypothesis testing (Fisher-Snedecor procedure [19]) with a 99% confidence level.

A last category of results deals with the deterioration of the performances as the prediction horizon increases. Figure 7 shows the evolution of the monthly-averaged MAPE for the two optimal model of each splitting strategy, namely M_{SL} and M_{TS}.

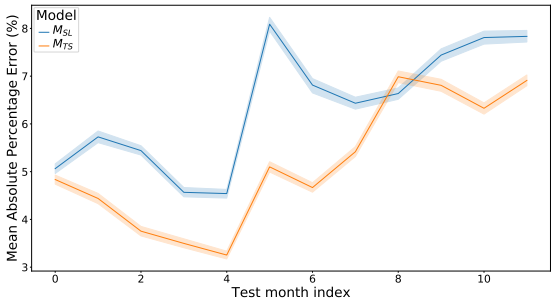

Fig. 7. Monthly Test MAPE for models M_{SL} and M_{TS}

The best model obtained from the temporal-split, M_{TS} presents a later and lower performance decrease as the prediction horizon decreases. In addition to a larger degradation from the first months, M_{SL} exhibits a larger confidence interval, hence more variance. This trend is confirmed by Figure 8 which displays monthly APE distributions instead of single monthly averages.

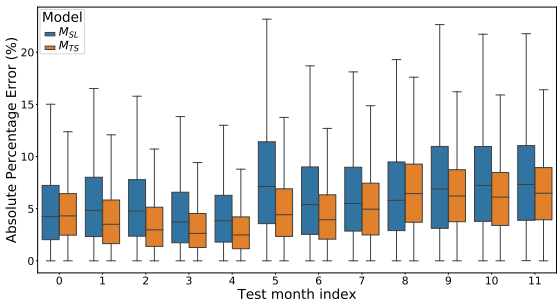

Fig. 8. Monthly Test APE for models M_{SB} and M_{TS}

These last two figures also show that after a given amount of time, the two models tend to harmonize towards higher error rates, which could be explained by the presence of additional, time-evolving factors such as hull fouling.

V. Conclusions and Perspectives

The aim of this paper was to provide new methodological insights into the problem of propulsion power prediction. After the identification of several knowledge gaps from the state of the art, we have proposed a systematic methodology in order to assess appropriate splitting strategies and model complexities for the task of predicting the propulsion power of cruise ships on unseen future data. The main conclusions can be summarized as follows:

- A shuffle-split including the test set, as shown in Figure 1, leads to the cross-talk problem, a methodological flaw

yielding an overestimation of the models performances (see Figure 2);
- Given an appropriate independent test set of unseen future data, a training methodology based on a temporal-split training and validation historical data set always outperforms a shuffle-split cross-validation. A shuffle split underestimates the validation error by overfitting during the training phase;
- For a given splitting approach, within the MLP framework, parsimonious models are equal or better than most refined architectures, with less variance in the test errors.

Several issues merit discussion:

1) Explanation vs. Prediction: A shuffle-split methodology on high frequency sensor data would be appropriate in situations where future prediction is not the final goal. If the goal is to analyze the effects of a certain variable on a historical dataset (reanalysis), such an approach may be appropriate. In these cases, ML is employed as an explanatory tool rather than a prediction tool, and could possibly be applied to evaluating factors affecting the propulsion efficiency (e.g., [4]) or evaluating the efficacy of certain equipment;
2) Even though the error increases for data outside the window of the training data, the temporal split model still performs with lower error at a 6-months horizon. This suggests that, even though we might expect the model to perform worse with time as new data is encountered and hull fouling may occur, the model trained using temporal split has generalized better than its counter part;
3) Sensitivity analysis on the effect of window's size on overfitting: It is likely that larger windows (e.g., 30, 60, 90 minutes) would be able to overcome cross-talk. An extreme case would be to shuffle entire trip segments (port-to-port) when enough data is available.

Perspective work to improve on the understanding and prediction of propulsion power for cruise ships could include a combination of supervised and unsupervised algorithms, and the inclusion of physical knowledge to the problem. Within the context of supervised learning, data could be segmented by trips and the trips could be shuffled and split into train, validation, and test sets. This would likely enable the learning of a more generalized model and avoid the cross-talk problem. Furthermore, temporal aspect can be considered a) using time-series based approaches (e.g., ARIMA models, LSTM); and b) by including time passed since hull/propeller cleaning as a feature to encapsulate the biofouling effect.

Unsupervised methods could be used to classify data based on power consumption (similar to [20], [21]) or power consumption deviation from an expected (theoretical) value. The latter could provide insights on factors affecting deviation from – for example – a physical model, and could be used to build a hybrid predictive model. Moreover, if entire trips are segmented, clustering could be performed at the trip-level considering consumed energy instead of power.

REFERENCES

[1] I. M. Organization, "Third IMO GHG Study 2014," https://www.imo.org/en/OurWork/Environment/Pages/Greenhouse-Gas-Studies-2014.aspx, accessed: 2020-06-30.

[2] ——, "IMO2020 – cutting sulphur oxide emissions," https://www.imo.org/en/MediaCentre/HotTopics/Pages/Sulphur-2020.aspx, accessed: 2020-07-15.

[3] F. Baldi, F. Ahlgren, T.-V. Nguyen, M. Thern, and K. Andersson, "Energy and exergy analysis of a cruise ship," *Energies*, vol. 11, no. 10, p. 2508, 2018.

[4] A. Laurie, E. Anderlini, J. Dietz, and G. Thomas, "Machine learning for shaft power prediction and analysis of fouling related performance deterioration," *Ocean Engineering*, p. 108886, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821003218

[5] B. P. Pedersen and J. Larsen, "Prediction of full-scale propulsion power using artificial neural networks," in *Proceedings of the 8th international conference on computer and IT applications in the maritime industries (COMPIT'09), Budapest, Hungary May*, 2009, pp. 10–12.

[6] J. P. Petersen, D. J. Jacobsen, and O. Winther, "Statistical modelling for ship propulsion efficiency," *Journal of Marine Science and Technology*, vol. 17, pp. 30–39, 3 2012.

[7] P. Theodoropoulos, C. C. Spandonidis, N. Themelis, C. Giordamlis, and S. Fassois, "Evaluation of different deep-learning models for the prediction of a ship's propulsion power," *Journal of Marine Science and Engineering*, vol. 9, no. 2, p. 116, 2021.

[8] Y.-R. Kim, M. Jung, and J.-B. Park, "Development of a fuel consumption prediction model based on machine learning using ship in-service data," *Journal of Marine Science and Engineering*, vol. 9, no. 2, p. 137, 2021.

[9] T. Uyanık, Çağlar Karatuğ, and Y. Arslanoğlu, "Machine learning approach to ship fuel consumption: A case of container vessel," *Transportation Research Part D: Transport and Environment*, vol. 84, p. 102389, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920920305769

[10] W. Mao, I. Rychlik, J. Wallin, and G. Storhaug, "Statistical models for the speed prediction of a container ship," *Ocean Engineering*, vol. 126, pp. 152–162, 11 2016.

[11] A. Brandsæter and E. Vanem, "Ship speed prediction based on full scale sensor measurements of shaft thrust and environmental conditions," *Ocean Engineering*, vol. 162, pp. 316–330, 8 2018.

[12] N. Themelis, C. Giordamlis, and C. Spandonidis, *Data acquisition and processing techniques for a novel performance monitoring system based on KPIs*, 2019. [Online]. Available: https://www.researchgate.net/publication/335856563

[13] Y. Du, Q. Meng, S. Wang, and H. Kuang, "Two-phase optimal solutions for ship speed and trim optimization over a voyage using voyage report data," *Transportation Research Part B: Methodological*, vol. 122, pp. 88–114, 4 2019.

[14] Y. B. Farag and A. I. Ölçer, "The development of a ship performance model in varying operating conditions based on ann and regression techniques," *Ocean Engineering*, vol. 198, p. 106972, 2 2020.

[15] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801819304561

[16] E. B. Beşikçi, O. Arslan, O. Turan, and A. I. Ölçer, "An artificial neural network based decision support system for energy efficient ship operations," *Computers and Operations Research*, vol. 66, pp. 393–401, 2 2016.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International journal of forecasting*, vol. 9, no. 4, pp. 527–529, 1993.

[19] J. Johnston and J. DiNardo, "Econometric methods," 1963.

[20] D. Brodić and A. Amelio, "Detecting of the extremely low frequency magnetic field ranges for laptop in normal operating condition or under stress," *Measurement*, vol. 91, pp. 318–341, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026322411630210X

[21] L. Ruiz, M. Pegalajar, R. Arcucci, and M. Molina-Solana, "A time-series clustering methodology for knowledge extraction in energy consumption data," *Expert Systems with Applications*, vol. 160, p. 113731, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420305558