Power Demand Forecasting for a Hybrid Marine Energy System with Shallow and Deep Learning

Walker, Jake M.; Coraddu, Andrea; Oneto, Luca

DOI
10.1109/OCEANS51537.2024.10682263

Publication date
2024

Document Version
Final published version

Published in
Proceedings OCEANS 2024 - Singapore, OCEANS 2024

Power Demand Forecasting for a Hybrid Marine Energy System with Shallow and Deep Learning

1st Jake M. Walker
Department of Maritime and Transport Technology
Delft University of Technology
Delft, The Netherlands
j.m.walker@tudelft.nl

2nd Andrea Coraddu
Department of Maritime and Transport Technology
Delft University of Technology
Delft, The Netherlands
a.coraddu@tudelft.nl

3rd Luca Oneto
Department of Informatics, Bioengineering, Robotics and Systems Engineering
Università Degli Studi di Genova
Genova, Italy
luca.oneto@unige.it

OCEANS 2024 - SINGAPORE | 979-8-3503-6207-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/OCEANS51537.2024.10682263

***Abstract*—To ensure that future autonomous surface ships sail in the most sustainable way, it is crucial to optimize the performance of the Energy and Power Management (EPM) system. However, marine EPM systems are complex and often coordinate various distributed energy resources, energy storage systems, and power grids to ensure reliable and safe power delivery. Traditional control methods for marine EPM systems are limited by evaluating processes using simplified component models over a short time horizon, or relying on historical insights gained from earlier journeys, and are not the optimal approach for complex hybrid marine EPM systems. Advanced control strategies, such as Model Predictive Control (MPC), offer a promising control method that considers predicted future system responses over an extended time horizon to determine the best control input, making them an effective strategy for optimizing the performance of hybrid marine EPM systems. However, to learn the onboard energy profiles based on component behavior in a hybrid system from past experiences is not a trivial task, and one of the primary barriers to implementing MPC for marine EPM control. For this reason, in this work, we address the challenge of learning energy profiles for a marine EPM system by utilizing shallow and deep machine learning for total power demand forecasting. The forecast is an essential reference for an MPC-based controller and will enable this control strategy to provide reliable and safe power delivery for hybrid marine EPM systems. The proposed approach compares state-of-the-art machine learning models to identify the best-performing algorithm, considering accuracy and computational requirements. We illustrate the potential of the proposed approach by using real world operational data from a vessel with a hybrid marine EPM system. Results indicate that shallow models, trained on engineered features handcrafted with classical signal processing techniques, allow forecasting the total power demand up to a horizon of 5min with minimal loss in accuracy and a negligible computational burden.**

I. Introduction

The global push to reduce emissions [1], [2] has led to the development of sophisticated marine energy systems. These systems integrate various clean energy sources, including liquefied natural gas [3], marine gas oil [4], low sulfur fuel oil [5], and hydrogen [6]. Additionally, they incorporate energy storage devices such as batteries and fuel cells [6], alongside conventional internal combustion engines. Consequently, the design, integration, and control of these hybrid energy systems have become increasingly complex [7]. This complexity renders it impractical to construct and test these systems before implementing necessary changes. Overcoming this challenge is critical for the sustainable deployment of autonomous surface ships [8], [9].

To enhance energy efficiency, component lifespan, and safety, various strategies are utilized to regulate marine energy systems [10]. However, implementing effective control strategies for hybrid Energy and Power Management (EPM) systems is not trivial [7]. These control strategies must include real-time feedback, state predictions, and forecasts of the system's future behavior to optimize control inputs [7], [11]–[13].

In fact, relying on classical control methods such as rule-based controllers [14], which rely on simplified component models, to control hybrid EPM systems is not an effective strategy given the complexity of the hybrid system. For this reason, alternative control strategies must be explored that are able to leverage highly accurate component models with minimal computational burden.

One approach to optimizing the control of EPM systems is to leverage advanced techniques like Model Predictive Control (MPC), which can incorporate advanced component models to accurately simulate the energy profile during EPM [15]. A key advantage of MPC strategies is that it is possible to forecast the system parameters over a set time horizon, which enhances the EPM performance compared to classical controllers that lack advanced knowledge of the energy profile [16].

However, to learn the onboard energy profiles based on component behavior in a hybrid system from past experiences is not a trivial task, and one of the primary barriers to implementing MPC MPC for marine EPM control. In fact, with an accurate model of the onboard energy profile it is possible to accurately forecasting the future power demand of a hybrid marine energy system and ensure safe and reliable power delivery. Currently, there are two main approaches when it comes to learning the onboard energy profiles. First, the conventional approach, is to model the vessel behavior using an understanding of the underlying physics of the problem [17] and optimize EPM systems based on these insights. However, modeling the physics of multiple components in a hybrid system, in real-time and forecasting the behavior, is challenging without sacrificing model accuracy due to necessary simplifications to reduce the computational demand. Alternatively, Machine Learning (ML) models, based solely on historical data, can learn the input-output behavior of a phenomenon without needing to model the underlying physics [18]. The advantage of ML models lies in their minimal computational burden upon deployment. However, developing ML models requires an extensive computational effort during the training phase. In the context of implementing MPC for hybrid EPM systems, ML models offer an effective solution for accurately forecasting future power system loads with minimal computational burden during deployment, which makes them well suited for real-time and forecasting applications. Nevertheless, the development of ML models depends on the existence of an automatic monitoring system and data storage onboard vessels.

According to Valchev et al. [19], new-built vessels are equipped with automatic monitoring systems as a standard feature, while older ships are increasingly being retrofitted with these systems. As a result, high-quality, high-frequency data is becoming more easily accessible [19]. The wide array of onboard sensors on vessels has enabled researchers in the maritime industry to employ various subsets of the available data features to develop ML forecasting models [20]. In this context, different shallow and deep ML techniques have been investigated for various applications. For instance, Coraddu et al. [13] used various regression models to predict vessel fuel consumption and trim optimization, while Walker et al. [21] and Valchev et al. [22] both employed Kernel Regularized Least Squares for short-term forecasting of vessel motion and performance, respectively. Walker et al. [23] compared different shallow and deep models for vessel short-term motions prediction under various loading conditions. Elatter et al. [24] and Ghelardoni et al. [25] utilized support vector machines for forecasting vessel load. Mehrzadi et al. [26] demonstrated the efficiency of Recurrent Neural Network (RNN) in predicting thruster power to counteract environmental disturbances for dynamic positioning applications.

However, the use of shallow and deep learning models for forecasting power demand for hybrid propulsion systems has yet to be explored. Shallow models, which are the classical family of ML models, typically rely on less data than deep models and work on the basis of first extracting simple, handcrafted features from the data and then applying traditional statistical techniques for prediction. Deep Learning models typically rely on a significant amount of data to build but can automatically learn complex features directly from the raw data. Both shallow and deep learning models have demonstrated strong performance in various time series forecasting tasks [23], [27], [28]. However, for both model types, it is important to ensure proper training, validation, and testing of the model using relevant data and considerations such as data quality, feature selection, model architecture, and hyperparameter tuning. Therefore, this study aims to develop an accurate power demand forecasting model for a vessel with hybrid propulsion systems. The models are trained on real-world operational data from a vessel and demonstrate that the proposed models can forecast the total power demand without requiring a priori knowledge of the system.

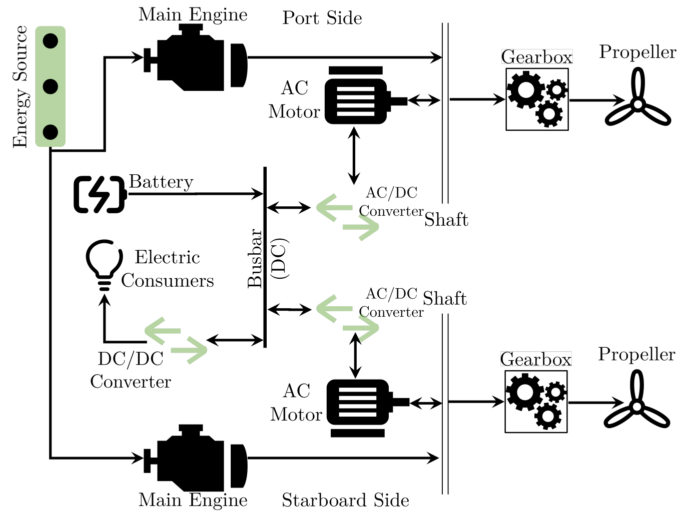


Fig. 1. The powertrain for the hybrid marine energy system under exam in this work.

The rest of the paper is organized as follows: Section II describes the powertrain and data leveraged in this work; Section III describes the method to build shallow and deep ML models for forecasting total power demand; Section IV presented the results of the method described in Section III on the data described in Section II; finally, Section V concludes the work.

II. Problem and Data Description

In this work we are concerned with forecasting the total power demand of a hybrid marine energy system using shallow and deep ML models. The powertrain for the hybrid marine energy system under exam in this work is shown in Figure 1.

Real world operational data for the described vessel was logged by the automatic monitoring system. The dataset features, presented in Table I, can be considered in 3 categories that represent the state of operation, the powertrain, and the power demand. For the problem at hand, namely forecasting the power demand, we consider the operational and powertrain features as the input variables; whereas, the target is the power features which is the power load demanded by the electric motors, batteries, and diesel engine. Although, for the problem at hand, we are actually interested in predicting the total power ($\mathrm{P_T}$), i.e., the combined power demanded by the electric motors, batteries, and diesel engines which is obtained by

$$\mathrm{P_T} = \mathrm{L_{DE,s}} + \mathrm{L_{DE,p}} + \mathrm{P_{EM,s}} + \mathrm{P_{EM,p}} + \mathrm{P_{BAT,s}} + \mathrm{P_{BAT,p}} \quad (1)$$

TABLE I

LIST OF FEATURES IN THE DATASET INCLUDING THE INPUTS AND OUTPUTS OF THE SHALLOW AND DEEP TIME-SERIES FORECASTING MODELS.

	Variable	Definition	Unit
Operational	t	Datetime	$[-]$
	v	Vessel Speed	[knots]
	ψ	Heading Position	[°]
	δ_{p}	Steering angle - Port	[°]
	δ_{s}	Steering angle - Starboard	[°]
Powertrain	$\mathrm{N_{EM,s}}$	Electric Motor speed - Starboard	[rpm]
	$\mathrm{N_{EM,p}}$	Electric Motor speed - Port	[rpm]
	$\mathrm{SoC_{BAT,p}}$	Battery State of Charge - Port	[%]
	$\mathrm{SoC_{BAT,s}}$	Battery State of Charge - Starboard	[%]
	$\mathrm{N_{DE,p}}$	Diesel Engine speed - Port	[rpm]
	$\mathrm{N_{DE,s}}$	Diesel Engine speed - Starboard	[rpm]
	$\mathrm{FC_{DE,p}}$	Diesel Engine fuel consumption - Port	[kg/min]
	$\mathrm{FC_{DE,s}}$	Diesel Engine fuel consumption - Starboard	[kg/min]
	$\mathrm{FC_{AUX}}$	Auxiliary generator fuel consumption	[kg/min]
	$\mathrm{N_{AUX}}$	Auxiliary generator engine speed	[rpm]
	$\mathrm{L_{AUX}}$	Auxiliary generator engine load	[kNm]
Power	$\mathrm{P_{EM,p}}$	Electric Motor power - Port	[kW]
	$\mathrm{P_{EM,s}}$	Electric Motor power - Starboard	[kW]
	$\mathrm{P_{BAT,p}}$	Battery power - Port	[kW]
	$\mathrm{P_{BAT,s}}$	Battery power - Starboard	[kW]
	$\mathrm{L_{DE,s}}$	Diesel Engine load - Starboard	[kW]
	$\mathrm{L_{DE,p}}$	Diesel Engine load - Port	[kW]

Note that, for confidentiality reasons, we will express the total power ($\mathrm{P_T}$) as a percentage of the total rated power, i.e., the sum of the rated power of the batteries, electric motors, and diesel engines.

For what concerns forecasting the total power demand for a hybrid EPM system, there are a couple of points worth discussing before we progress. The first point is that we do not consider forecasting times of extreme power use, such as during firefighting or rescue operations, because the control of the EPM system is not important compared to the vessel's primary function. The second point is that we do not forecast the auxiliary power loads (i.e., power for the lights, heating, towing equipment, or firefighting equipment) due to the unpredictable nature of human influence over these components. Considering these points, the available data was comprised of approximately 5×10^6 samples, recorded at a sampling rate of 5 seconds, over one year of operation.

III. METHOD

Forecasting the total power demand for a hybrid marine EPM system from historical data containing the input-output relationship is a supervised learning problem. We could consider a classical regression problem characterized by an input space $\mathcal{X} \subseteq \mathbb{R}^m$ (i.e., the operational and powertrain features of Table IV) and an output space $\mathcal{Y} \subseteq \mathbb{R}^b$ (i.e., the power features of Table IV). However, we are actually interested in forecasting the total power demand based on past values, which necessitates changing the classical framework in two ways. First, the input space is now composed of all the past information during the time frame $[t - \Delta^-, t]$ (i.e., $[\mathcal{X}, \mathcal{Y}] \subseteq \mathbb{R}^{m+b}$), hereinafter $\mathcal{X}^\Delta$, where the hyperparameter Δ^- represents the duration of historical data we want to include in the model. Second, the output space is now defined as the original target features in $\mathcal{Y}$ just at the time horizon $t + \Delta^+$, hereinafter $\mathcal{Y}^\Delta$, where the hyperparameter Δ^+ represents the prediction horizon. It is worth mentioning that balancing the effects of the temporal hyperparameters (Δ^- and Δ^+) must be considered carefully. Due to the curse of dimensionality, increasing Δ^- (which saturates the model with past information) can be problematic, while reducing Δ^- (and failing to capture the dynamics of the problem) limits our ability to make forecast accurately into the future [29]–[31]. The most suitable selection for Δ^+ relies on the problem at hand. While forecasting further ahead is typically the ideal scenario, extending the time horizon tends to lead to a decrease in prediction accuracy [29]–[31].

Using the proposed time-series forecasting regression framework, the goal is to estimate the unknown rule μ which maps $X \in \mathcal{X}^\Delta$ to $Y \in \mathcal{Y}^\Delta$ [29] by leveraging historical data (i.e., a dataset $\mathcal{D}_n : \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$.

In this work, we will approximate μ based on $\mathcal{D}_n$ using an ML algorithm $\mathscr{A}$. A particular algorithm $\mathscr{A}_{\mathcal{H}}$, characterized by its hyperparameters $\mathcal{H}$, maps $\mathcal{D}_n$ into a model f inside a space of possible ones $\mathcal{F}$. For the reasons described in Section I, we will consider both the shallow [29] and deep [32] families of algorithms. The difference between these families relies on how the input space $\mathcal{X}^\Delta$ is considered. For shallow learning, we start by manually transforming the dataset $\mathcal{X}^\Delta$, by implicit methods such as the kernel trick [33] or by manual features engineering [34], to create a representation vector $\phi(X) \in \mathbb{R}^d$. This representation captures essential information while discarding irrelevant details [35], [36]. In contrast, deep learning uses neural networks to automatically generate the representation $\phi(X) \in \mathbb{R}^d$ without explicitly defining it [32].

Regarding which machine learning algorithm to employ for this application, we follow the principles of the no-free-lunch theorem [37], which necessitates testing multiple algorithms to find the best one for the task at hand. With this principle in mind, we selected to test three state-of-the-art shallow algorithms [33], [38], [39] and one deep one [32]. For the shallow, we picked algorithms coming from two different families. In the first family, Kernel Methods [33], we employed a ridge regression model where the main idea is that the data is first mapped into a representation space and a linear solution to the problem is found. We considered KERNel ridge regression (KERN) with a Gaussian kernel for the reasons described in [40]. In the second family, Ensemble Methods [41]–[43], we selected to test two algorithms based on fundamentally different techniques to build the ensembles. First, we employed the BAGging technique (BAG) [41], [42] which is concerned with randomly sampling a subset of the training data to build different trees and averaging the models' outputs to reduce variance and improve performance. Subsequently, we opted to try a BOOsting technique (BOO) [43] which, in contrast, sequentially builds models, where each new tree attempts to correct the errors made by the previous ones. This technique gradually leads to improved model performance.

For the shallow algorithms, we considered the following hyperparameters

- KERN required tuning both the regularization hyperparameter C and the kernel coefficient γ
- BAG requires tuning the number of features randomly sampled from the entire set of features at each node n_f and the maximum number of elements in each leaf n_l. Since the performance of ensemble methods improves with the number of trees n_t, we capped n_t at 1000 for computational tractability;
- BOO required tuning the gradient learning rate l_r, the maximum depth of each tree n_d, the minimum loss reduction m_l, the number of points to randomly sample from the entire training set for each tree creation n_b, and the number of features to randomly sample from the entire set of features at each node n_f.

In addition to exploring shallow models, we have taken inspiration from research demonstrating the superior performance of deep learning models in various applications, particularly for the problem under exam, namely, time-series forecasting. Given the complexities and specific characteristics of our forecasting problem, it is crucial to leverage recent advancements in deep learning for this domain. While architectures such as the Long-Short Term Memory (LSTM) network and Bidirectional-LSTM network have performed well on certain time-series forecasting problems, it has been documented that these models suffer when dealing with problems with multiple temporal scales [44]. To overcome this issues, the deep temporal convolutional network (TCN) [44] is one of the most promising state-of-the-art deep learning architectures currently demonstrated in the literature and is proven to have performed well on other forecasting problems in the maritime domain [23]. For the TCN, the following hyperparameters need to be tuned: the learning rate l_r, the dropout rate $d_{r,0}$ of each TCN layer and the last layer, the regularization coefficient C, the number of TCN blocks h_l, the number of filters on each block n_i, and the kernel size for each series and block $k_{s,i}$. The TCN was implemented with custom software leveraging for the TensorFlow [45] Python module.

A summary of the hyperparmeters and the associated search space for each of the shallow and deep algorithms is reported in Table II.

The problem we still have to address is how to select the best hyperparameters for each of the algorithms and estimate the performance of the final model. Model Selection (MS) and Error Estimation (EE) deal exactly with these problems [30]. We will rely on a resampling method to split the original dataset $\mathcal{D}_n$ many (n_r) times into three independent datasets called learning, validation and test sets, respectively $\mathcal{L}_l^r$, $\mathcal{V}_v^r$,

TABLE II
MODEL HYPERPARAMETERS AND ASSOCIATED HYPERPARAMETER SPACE FOR EACH ALGORITHM TESTED IN THIS WORK.

Algorithm		Hyperparameters
Shallow	KERN	γ: $\{0.1, 0.01, 0.001, 0.0001\}$ $C : \{0.001, 0.01, 0.1, 1, 10, 100\}$
	BAG	$n_f : \{d^{1/3}, d^{1/2}, d^{3/4}\}$ $n_l : \{1, 3, 5, 10\}$ $n_t : \{1000\}$
	BOO	$l_r : \{0.01, 0.02, 0.03, 0.04, 0.05\}$ $n_d : \{3, 5, 10\}$ $m_l : \{0, 0.1, 0.2\}$ $n_b : \{0.6n, 0.8n, 1n\}$ $n_f : \{0.5d, 0.8d, 1d\}$
Deep	TCN	l_r: $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$ $d_{r,0} : \{0.1, 0.15, \ldots, 0.5\}$ $C : \{0.00001, 0.00005, 0.000001\}$ $h_l : \{1, 2, 3, 4\}$ $n_i : \{16, 32, 64, 128, 256\}$ $k_{s,i} : \{3, 5, 7, 9, 11\}$

and $\mathcal{T}_t^r$, with $r \in \{1, \cdots, n_r\}$ such that

$$\mathcal{L}_l^r \cap \mathcal{V}_v^r = \oslash, \quad \mathcal{L}_l^r \cap \mathcal{T}_t^r = \oslash, \quad \mathcal{V}_v^r \cap \mathcal{T}_t^r = \oslash \tag{2}$$

$$\mathcal{L}_l^r \cup \mathcal{V}_v^r \cup \mathcal{T}_t^r = \mathcal{D}_n \tag{3}$$

During the MS, we want to find the best combination of hyperparameters (in the set of possible ones: see Table II) for a shallow and deep algorithm $\mathscr{A}_{\mathcal{H}}$, which corresponds to

$$\mathcal{H}^* : \quad \arg\min_{\mathcal{H} \in \mathfrak{H}} \sum_{r=1}^{n_r} M(\mathscr{A}_{\mathcal{H}}(\mathcal{L}_l^r), \mathcal{V}_v^r), \tag{4}$$

where a model $f = \mathscr{A}_{\mathcal{H}}(\mathcal{L}_l^r)$ is developed with a shallow or deep algorithm $\mathscr{A}$, with the hyperparameters $\mathcal{H}$, and with the data $\mathcal{L}_l^r$, and where $M(f, \mathcal{V}_v^r)$ is an error metric which represents how well the model approximates the real phenomenon. The main idea behind resampling is that since the data in $\mathcal{L}_l^r$ are independent from the data in $\mathcal{V}_v^r$, $\mathcal{H}^*$ should is the combination of hyperparameters which results in the smallest error on data independent from the training set. Then, during the EE, we want to evaluate the performance of the final model $f^*_{\mathscr{A}} = \mathscr{A}_{\mathcal{H}^*}(\mathcal{D}_n)$ by evaluating

$$M(f^*_{\mathscr{A}}) = \frac{1}{n_r} \sum_{r=1}^{n_r} M(\mathscr{A}_{\mathcal{H}^*}(\mathcal{L}_l^r \cup \mathcal{V}_v^r), \mathcal{T}_t^r). \tag{5}$$

Similarly to before, because the data in $\mathcal{L}_l^r \cup \mathcal{V}_v^r$ are independent from $\mathcal{T}_t^r$, $M(f^*_{\mathscr{A}})$ is an unbiased estimator of the true performance of the final model according to the metric M [30]. One important note is that, because of the dependence in time between the samples, there is an additional constraint when resampling $\mathcal{D}_n$ where we preserve the continuity in time for $\mathcal{L}_l^r$, $\mathcal{V}_v^r$, and $\mathcal{T}_t^r$ [46].

Finally, when it comes to the metric M that we will use to represent how well the model approximates the real phenomenon, we have selected to report the Relative Error in Percentage (REP) which quantifies the average relative difference between the prediction ($f(X_i^r)$) and the actual value (Y_i^r) as a percentage

$$\text{REP}(f) = \frac{100}{n_r} \sum_{i=1}^{n_r} \frac{|f(X_i^r) - Y_i^r|}{Y_i^r} \tag{6}$$

For the particular problem at hand, it is not reasonable to synthesize the quality of the predictors based on purely numerical metrics, so we will also rely on a visualization of the track in time between the real and predicted values.

IV. Results

This section presents the results of the method described in Section III using the data described in Section II on the problem described in the very same section, namely, forecasting the total power demand of a hybrid marine energy system.

For all of the experiments in this work, we considered temporal hyperparameters based on the problem at hand, i.e., a forecast horizon in the range of 1–5 minutes to prescribe optimal control of a marine EPM system (see Section I).

$$\begin{aligned} \Delta^+ \in &\quad \{1\text{min}, 2\text{min}, 5\text{min}\} \\ \Delta^- \in &\quad \{5\text{s}, 30\text{s}, 1\text{min}, 5\text{min}, 10\text{min}\} \end{aligned}$$

Additionally, we have performed the resampling (n_r) 30 times and reported the error metrics with the interval of confidence evaluated using the t-student's distribution (with $n_r - 1$ degrees of freedom) at 95% confidence.

Regarding the computational hardware utilized in this work, experiments with shallow models were conducted using 2×Intel XEON E5-6248R 24C 3.0GHz CPUs and 192GB of memory; while experiments with deep models were supported by 2×NVIDIA Tesla K80 GPUs with 24GB of memory.

To test the proposed method for forecasting the total power of a hybrid marine powertrain, in the first instance, we used the raw signal from the vessel automatic monitoring system (without additional features engineering) as the input to the four different shallow and deep ML algorithms. Table III presents the Total Power (P_T) REP [%] for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each algorithm trained on the raw signal from the vessel automatic monitoring system (without additional features engineering).

It is worth making a number of observations based on Table III

- With the exception of the KERN algorithm at the 1min horizon, all algorithms (shallow and deep), exhibit similar performance across different time horizons;
- In terms of computational demands during the model training phase, the deep model is more resource-intensive compared to the shallow models;
- Both shallow and deep models have minimal computational requirements during the forward phase (Test) of their operations;
- As the time horizon extends further into the future, there is a noticeable increase in the value of the REP.

TABLE III
Total Power (P_T) REP [%] for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each algorithm trained on the raw signal.

Algorithm		REP [%] for different Δ^+			Time	
		1min	2min	5min	Train [h]	Test [μs]
Shallow	KERN	7.5 ± 1.2	9.9 ± 1.6	19.9 ± 4.3	0.7 ± 0.2	10.5 ± 1.0
	BAG	10.8 ± 0.6	11.1 ± 1.4	21.6 ± 3.5	3.2 ± 0.1	5.7 ± 1.2
	BOO	10.3 ± 0.4	10.8 ± 1.2	22.3 ± 3.6	5.9 ± 0.4	8.4 ± 0.6
Deep	TCN	10.2 ± 0.2	11.0 ± 0.4	12.4 ± 0.5	6.5 ± 0.6	17.0 ± 6.3

At first glance, the results of Table III appear to be in line with the expectations laid out in Sections I–III. However, contrary to the literature on the topic, these results suggest that the deep model did not automatically derive a representation from the raw signal that could outperform a shallow model trained on the raw signal directly. There are a couple of explanations to why this was the case. First, there can be too few data for the deep model to automatically learn a representation better than the raw signal itself (which explains why the performance of the shallow and deep models are similar). Next, it can be the case that the selected hyperparameter ranges were not suitable for the particular problem under consideration; however, the total time allocated to running deep experiments was in excess of 8 days (see Table III) so it is computationally prohibitive to repeat the experiment using different hyperparameter ranges.

Due to the limitations of the first experiment, and to further test the proposed method for forecasting the total power of a hybrid marine powertrain, in the second instance, we engineered a handcrafted set of features from the raw signal based on classical signal processing techniques as the input to the three different shallow ML algorithms. A description of engineered features and their symbols is presented in Table IV. Table V presents the Total Power (P_T) REP [%] for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each shallow algorithm trained on the engineered features.

Now, there are a few important observations to make

- When trained on engineered features, all of the shallow algorithms consistently yield better results compared to their performance when trained directly on the raw signal across various forecast horizons (see Figure 2);
- Shallow models, when trained on engineered features, exhibit slightly higher computational requirements during the forward phase (Test) compared to those trained directly on the raw signal;
- Similarly to before, as the time horizon extends further into the future, there is a noticeable increase in the value of the REP.

The results of Table V indicate, that for this particular problem and data, the shallow models trained on engineered features are the best approach to forecast the total power of

TABLE IV
DESCRIPTION OF ENGINEERED FEATURES AND THEIR SYMBOLS.

Function	Description
mean	Mean value
var	Variance
mad	Median absolute value
max	Largest value in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of squares
iqr	Interquartile range
entropy	Signal Entropy
correlation	Correlation coefficient between series
kurtosis	Signal Kurtosis
skewness	Signal Skewness
maxFreqInd	Largest frequency component
argMaxFreqInd	Index largest frequency component
meanFreq	Frequency signal weighted average
skewnessFreq	Frequency signal Skewness
kurtosisFreq	Frequency signal Kurtosis
ampSprec	Amplitude Spectrum of the frequency signal
angle	Phase angle of the frequency signal

TABLE V
TOTAL POWER ($\mathrm{P_T}$) REP [%] FOR DIFFERENT FORECAST HORIZONS (Δ^+) WITH THE OPTIMAL MODEL AND TEMPORAL (Δ^-) HYPERPARAMETERS FOR EACH SHALLOW ALGORITHM TRAINED ON THE ENGINEERED FEATURES.

Algorithm		REP [%] for different Δ^+			Time	
		1min	2min	5min	Train [h]	Test [ms]
Shallow	KERN	3.7 ± 0.8	4.9 ± 0.5	9.2 ± 1.6	0.8 ± 0.2	14.7 ± 1.9
	BAG	3.7 ± 0.6	4.3 ± 0.6	9.4 ± 1.4	3.2 ± 0.2	9.0 ± 0.6
	BOO	3.5 ± 0.5	4.2 ± 0.5	9.3 ± 1.1	6.0 ± 0.3	10.2 ± 1.5

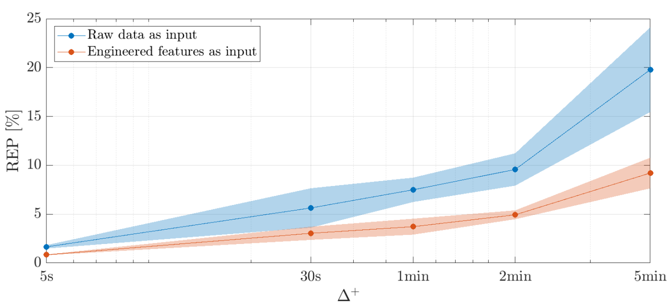
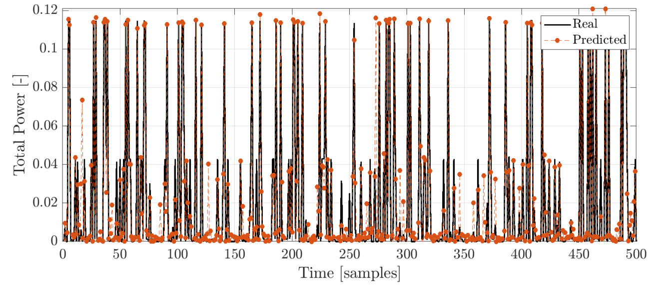


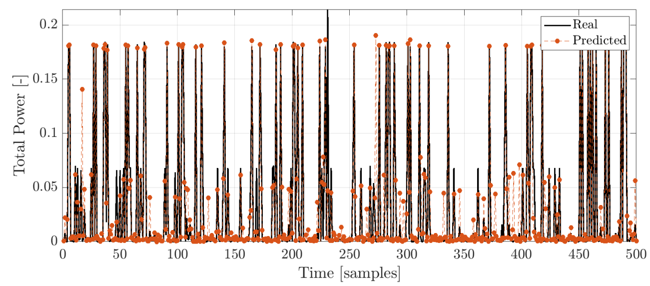
Fig. 2. Best shallow algorithm performance (REP[%]) when trained on the raw data as input (Table III) and the engineered features as input (Table V) across various forecast horizons.

a hybrid marine powertrain. However, because it is quite hard to synthesize the results of regression problems from only numerical quantifiers, we have decided to present a visual metric alongside the previous results. Figure 3 presents a portion of the track-in-time (500 samples) for the best shallow algorithm for each of the different forecast horizons according to Table V. The results of Table V are well reflected in Figure 3a– 3c through the following observations

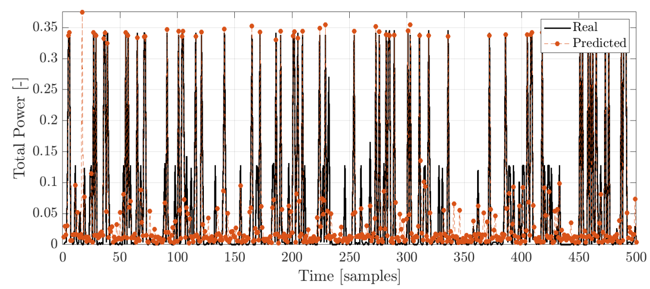
- In the first instance ($\Delta^+ = 1\mathrm{min}$), the model is able to well represent the signal and the real peaks are well matched by the predictions (see Figure 3a);
- In the second instance ($\Delta^+ = 2\mathrm{min}$), the model is slightly worse at matching the magnitude of the real peaks with its predictions (see Figure 3b);
- In the final instance ($\Delta^+ = 5\mathrm{min}$), the model is not able to predict as many of the peaks in the signal (see Figure 3c);



(a) $\Delta^+ = 1\mathrm{min}$ - BOO



(b) $\Delta^+ = 2\mathrm{min}$ - BOO



(c) $\Delta^+ = 5\mathrm{min}$ - KERN

Fig. 3. A portion of the track-in-time for the best shallow algorithm for each of the different forecast horizons according to Table V.

It can be observed that . In the

V. CONCLUSION

Optimizing the Energy and Power Management (EPM) systems is vital for the sustainable deployment of autonomous surface ships. The complexity of marine EPM systems presents significant challenges in design, simulation, and control. This is particularly true when coordinating various distributed energy resources and power grids to ensure reliable power delivery. To tackle these challenges, one promising approach is the integration of Model Predictive Control (MPC) with Machine Learning (ML) models. MPC strategies, utilizing forecasts of system parameters, can significantly improve performance compared to traditional controllers.

For this reason, this study introduces a framework to learn the energy profile of a hybrid marine energy system. We show that, by employing both shallow and deep ML models, we can develop an accurate forecast of the total power demand over a 1–$5\mathrm{min}$ forecast horizon. Initially, using the raw data as input to shallow and deep ML models yielded moderate accuracy (10–20% REP over a 1–$5\mathrm{min}$ forecast horizon). However, incorporating engineered features based on classical signal processing significantly enhanced model performance (3–10% REP over a the very same forecast horizons).

Our study focused on data from a single vessel over a limited time frame and a necessity for wider validation still exists. The performance of deep learning models, in particular, may improve with more extensive data. A crucial next step for this research is to integrate the forecasting model into an MPC-based controller in a simulation environment. With this in mind, we observed that the proposed ML models, both shallow and deep, impose minimal computational burden (milliseconds), making them suitable for real-time and forecasting applications for hybrid marine energy systems control.

ACKNOWLEDGMENT

All experiments have been run on the DelftBlue supercomputer at the Delft High-Performance Computing Center [47], which hosts 238 Compute nodes with a total of 476 Intel XEON E5-6248R 24C 3.0GHz CPUs and 192 GB of Memory per node.

This research is supported by the *Sustainable Hydrogen Integrated Propulsion Drives* (SH2IPDRIVE) project, which has received funding from RvO (reference number MOB21013), through the RDM regulation of the Ministry of Economic Affairs and Climate Policy.

This publication is part of the project SEANERGETIC (with project number KICH1.KICH1.21.003) of the research program Zero Emission and Circular Shipping (KIC) which is financed by the Dutch Research Council (NWO).

REFERENCES

[1] C. Dere and C. Deniz. Load Optimization of Central Cooling System Pumps of a Container Ship for the Slow Steaming Conditions to Enhance the Energy Efficiency. *Journal of Cleaner Production*, 222:206–217, 2019.

[2] UNCTAD. *Review of Maritime Transports*. United Nations, New York and Geneva, 2013.

[3] N. Pavlenko, B. Comer, Y. Zhou, N. Clark, and D. Rutherford. The Climate Implications of using LNG as a Marine Fuel. *Swedish Environmental Protection Agency: Stockholm, Sweden*, 2020.

[4] C. Ji and M.M. El-Halwagi. A Data-Driven Study of IMO Compliant Fuel Emissions with consideration of Black Carbon Aerosols. *Ocean Engineering*, 218:108241, 2020.

[5] B. Stolz, M. Held, G. Georges, and K. Boulouchos. Techno-economic analysis of renewable fuels for ships carrying bulk cargo in Europe. *Nature Energy*, 7(2):203–212, 2022.

[6] H. Xing, C. Stuart, S. Spence, and H. Chen. Alternative fuel options for low carbon maritime transportation: Pathways to 2050. *Journal of Cleaner Production*, 297:126651, 2021.

[7] C. Wei, X. Bai, and T. Kim. Advanced control and optimization for complex energy systems, 2020.

[8] E. Jokioinen, J. Poikonen, M. Hyvönen, A. Kolu, T. Jokela, J. Tissari, A. Paasio, H. Ringbom, F. Collin, M. Viljanen, R. Jalonen, R. Tuominen, M. Wahlström, J. Saarni, S. Nordgerg-Davies, and H. Makkonen. Remote and Autonomous Ships - The next step. *AAWA: Advanced Autonomous Waterborne Applications*, pages 1–87, 2016.

[9] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J.M. Vallet García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Zahidul H. Bhuiyan, S. Särkkä, and V.V. Lehtola. Sensors and ai techniques for situational awareness in autonomous ships: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):64–83, 2022.

[10] P. Xie, J.M. Guerrero, S. Tan, N. Bazmohammadi, J.C. Vasquez, M. Mehrzadi, and Y. Al-Turki. Optimization-based power and energy management system in shipboard microgrid: A review. *IEEE systems journal*, 2021.

[11] P. Wu and R. Bucknall. Hybrid fuel cell and battery propulsion system modelling and multi-objective optimisation for a coastal ferry. *International journal of hydrogen energy*, 45(4):3193–3208, 2020.

[12] K. Wang, X. Yan, Y. Yuan, X. Jiang, X. Lin, and R. R. Negenborn. Dynamic optimization of ship energy efficiency considering time-varying environmental factors. *Transportation Research Part D: Transport and Environment*, 62:685–698, 2018.

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

[14] J. Han, J. Charpentier, and T. Tang. An energy management system of a fuel cell/battery hybrid boat. *Energies*, 7(5):2799–2820, 2014.

[15] P. Doganis, E. Aggelogiannaki, and H. Sarimveis. A combined model predictive control and time series forecasting framework for production-inventory systems. *International Journal of Production Research*, 46(24):6841–6853, 2008.

[16] S. Antonopoulos, K. Visser, M. Kalikatzarakis, and V. Reppa. MPC Framework for the Energy Management of Hybrid Ships with an Energy Storage System. *Journal of Marine Science and Engineering*, 9(9):993, 2021.

[17] A. Coraddu, M. Kalikatzarakis, G. Theotokatos, R. Geertsma, and L. Oneto. Physical and data-driven models hybridisation for modelling the dynamic state of a four-stroke marine diesel engine. In *Engine Modeling and Simulation*, pages 145–193. Springer, 2021.

[18] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[19] I. Valchev, A. Coraddu, M. Kalikatzarakis, R. Geertsma, and L. Oneto. Numerical methods for monitoring and evaluating the biofouling state and effects on vessels' hull and propeller performance: A review. *Ocean Engineering*, 251:110883, 2022.

[20] M. Üney, L. M. Millefiori, and P. Braca. Data driven vessel trajectory forecasting using stochastic generative models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8459–8463. IEEE, 2019.

[21] J.M. Walker, A. Coraddu, V. Garofano, and L. Oneto. Artificial Intelligence Based Short-Term Motions Forecasting for Autonomous Marine Vehicles Control. In *Proceedings of the International Ship Control Systems Symposium*, volume 16, page 35, 2022.

[22] I. Valchev, A. Coraddu, L. Oneto, M. Kalikatzarakis, W. Tiddens, and R.D. Geertsma. Artificial Intelligence-based short-term forecasting of vessel performance parameters. In *Proceedings of the International Ship Control Systems Symposium*, volume 16, page 31, 2022.

[23] J.M. Walker, A. Coraddu, S. Savio, and L. Oneto. Shallow and Deep Learning Models for Vessel Motions Forecasting during Adverse Weather Conditions. *Modelling and Optimisation of Ship Energy Systems 2023*, Dec. 2023.

[24] E.E. Elattar, J. Goulermas, and Q.H. Wu. Electric load forecasting based on locally weighted support vector regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(4):438–447, 2010.

[25] L. Ghelardoni, A. Ghio, and D. Anguita. Energy load forecasting using empirical mode decomposition and support vector regression. *IEEE Transactions on Smart Grid*, 4(1):549–556, 2013.

[26] M. Mehrzadi, Y. Terriche, C. Su, P. Xie, N. Bazmohammadi, M. N. Costa, C. Liao, J. C. Vasquez, and J. M. Guerrero. A deep learning method for short-term dynamic positioning load forecasting in maritime microgrids. *Applied Sciences*, 10(14):4889, 2020.

[27] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[28] M. Sangiorgio and F. Dercole. Robustness of lstm neural networks for multi-step forecasting of chaotic time series. *Chaos, Solitons & Fractals*, 139:110045, 2020.

[29] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[30] L. Oneto. *Model selection and error estimation in a nutshell*. Springer, 2020.
[31] J.D. Hamilton. Time series analysis (Vol. 2). *Princeton, Princeton*, 1990.
[32] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
[33] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
[34] F. Cipollini, L. Oneto, A. Coraddu, S. Savio, and D. Anguita. Unintrusive monitoring of induction motors bearings via deep learning on stator currents. *Procedia computer science*, 144:42–51, 2018.
[35] M. Kuhn and K. Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.
[36] P. Duboue. *The art of feature engineering: essentials for machine learning*. Cambridge University Press, 2020.
[37] S.P. Adam, S.A.N. Alexandropoulos, P.M. Pardalos, and M.N. Vrahatis. No free lunch theorem: A review. *Approximation and optimization*, pages 57–82, 2019.
[38] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.
[39] M. Wainberg, B. Alipanahi, and B.J. Frey. Are random forests truly the best classifiers? *The Journal of Machine Learning Research*, 17(1):3837–3841, 2016.
[40] S.S. Keerthi and C.J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.
[41] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
[42] I. Orlandi, L. Oneto, and D. Anguita. Random forests model selection. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.
[43] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.
[44] V. D'Amato, L. Oneto, A. Camurri, and D. Anguita. The importance of multiple temporal scales in motion recognition: when shallow model can support deep multi scale models. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–10, 2022.
[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, 2015.
[46] J. D. Hamilton. *Time series analysis*. Princeton university press, 1994.
[47] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 1). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.