

Shallow and Deep Learning Models for Vessel Motions Forecasting during Adverse Weather Conditions

Jake M. Walker^a, Andrea Coraddu^{a,*}, Stefano Savio^b, and Luca Oneto^b

^aDelft University of Technology, Delft, Netherlands

^bUniversity of Genoa, Genoa, Italy

*a.coraddu@tudelft.nl

Abstract

Accurately forecasting vessel motions is a critical step towards achieving fast and accurate intelligent vessel control systems. Intelligent vessel control relies on accurate predictions of vessel motion to make informed decisions regarding control, maneuvering, and positioning, particularly during times of exogenous loading caused by adverse weather conditions. Hence, by accurately forecasting vessel motion accurately, the control system can anticipate potential issues (i.e., excessive trim or roll) and prescribe corrective actions before they become problematic. In this study, the authors propose two approaches to address the problem of vessel motion forecasting. The first approach relies on classical shallow learning models, whereas the second approach involves the use of state-of-the-art deep learning models for improved accuracy at further forecast horizons. Unlike shallow models, deep models can learn the required features directly from the data and therefore do not require *a priori* knowledge or additional features engineering. By leveraging deep learning models, the authors show that vessel motions can be forecasted further into the future without a significant loss in accuracy, thereby improving the overall effectiveness of the intelligent vessel control system. To support their statements, the authors use real operational data and compare the performance of the shallow and deep learning models. The results show that deep learning outperforms shallow learning models in terms of accuracy without a significant increase in the computational demand. Additionally, the authors demonstrate that their models remain accurate even under adverse weather conditions, indicating that they have practical applicability for vessel motions forecasting and can potentially improve the overall effectiveness of intelligent vessel control systems.

Keywords: Autonomous Vessels; Intelligent Control; State Prediction; Time-Series Forecasting, Signal Processing, Supervised Learning, Shallow Models, Deep Models.

1 INTRODUCTION

Intelligent and autonomous vessels have been proposed as an important step towards mitigating emissions from shipping, alleviating seafarers' fatigue, and enhancing safety measures at sea [1]. Nonetheless, the deployment of fully autonomous vessels across intricate mission scenarios remains a formidable challenge [2], [3]. According to the International Maritime Organization (IMO), achieving full autonomy for a vessel requires a control system capable of independently determining the most optimal course of action [4]. Therefore, prior to the deployment of fully functional autonomous vessels, concerted efforts must be directed towards developing intelligent control systems. Although numerous motion control systems have been documented in the literature [5], substantial work remains to bridge the gap between the predicted behavior and the actual responses of vessels to their surroundings [6].

To consider the behavior and dynamics of a ves-

sel in six Degrees of Freedom (DoF), contemporary physics-based models can be leveraged [7], which account for three translational motions (surge, sway, and heave) as well as three rotational motions (roll, pitch, and yaw) and are often characterized by a high accuracy and interoperability. Real-time solutions provided by numerical models such as [7] are a necessary step towards intelligent control systems and can accurately describe the state of the vessel while incorporating external disturbances, and determine the optimal force distributions needed to meet mission criteria. However, vessel control systems frequently operate under challenging conditions and intricate mission environments such as close proximity situations, densely populated areas, station keeping, mooring, automatic docking, and helicopter operations.

Irrespective of the application, for fully autonomous vessels to operate independently and prescribe the optimal course of action even in situations of high exogenous loading (e.g., high wind

speeds and sea swells). It is essential for intelligent control to accurately predict the short-term future state of the vessel (i.e., forecast) and not just at the present moment (i.e., nowcast) during these conditions. In the context of intelligent vessel control systems, state prediction facilitates protocols such as model predictive control where the horizon prediction is coupled with the real-time solution of a multiobjective optimization problem [6]. Hence, within the intelligent vessel control framework, the control system can anticipate potential issues (i.e., excessive trim or roll) due to short-term motions forecasting over a sufficiently large horizon and prescribe corrective actions (i.e., optimal force distributions) before they become problematic.

Existing control systems often rely on observers and state predictors such as the Kalman Filter [8]. However, real-time implementation issues frequently arise because of disparities between actual and forecasted motion attributed to simplified vessel or environment models. To address this disparity, contemporary sensor technology data can be combined with automatic control systems to enhance autonomous operations, assess mission feasibility, and determine optimal control strategies to ensure mission success.

Additionally, the complexity of the optimization problem increases with the forecast horizon (i.e., extending further into the future), which results in the computational complexity of physics-based solutions increasing significantly. Therefore, state prediction will benefit from a fast, novel, and accurate solution that leverages state-of-the-art machine learning models. For these reasons, this paper centers its focus on Shallow and Deep Learning models for vessel motions forecasting, which is a pivotal element for formulating intelligent control strategies and realizing the potential of fully autonomous intelligent vessel control systems. Additionally, this study demonstrates how these models can be developed to remain reliable even during adverse weather conditions characterized by periods of high exogenous loading.

The rest of this paper is organized as follows: Section 2 presents an overview of related work on machine learning models for vessel motions forecasting, Section 3 describes the problem at hand and the available data, Section 4 presents the proposed methodology, Section 5 outlines the results, and finally, Section 6 concludes the work.

2 RELATED WORK

For the sake of completeness, the authors have reported the current state-of-the-art approaches to vessel motion prediction using machine learning in this section with a tabulated summary of the related works found in Table 1.

In [8], the authors developed a model that yielded satisfactory performance in forecasting heave motion, with a forecast horizon of 15-30 [s] and a Root Mean Square Error (RMSE) between $\sim 0.05 - 0.2$ [m]. Additionally, the prediction of pitch and roll motion up to a 50s horizon was executed using a hybrid Nonlinear AutoRegressive (NAR) wavelet framework. The models displayed an accuracy of RMSE ~ 0.05 [$^{\circ}$] for pitch and RMSE ~ 0.13 [$^{\circ}$] for roll. Despite the promising results, they were derived from a rather limited dataset (650 samples), providing only a single day's motion description for an inertial platform.

In [9], the authors predicted the roll motion of a floating production unit using an Autoregressive Integrated Moving Average (ARIMA) deep learning model. They used of synthetic data captured at an extremely high frequency (15 Hz), and a hybrid model with a forecast range of 3-16 [s] resulted in an RMSE of ~ 0.04 [$^{\circ}$].

In [10], the authors conducted a study where they forecasted the heading angle with a horizon of 1s, resulting in an RMSE of ~ 0.2 [$^{\circ}$]. The authors leveraged a time-delay wavelet Neural Network (NN). When the horizon was extended to 3 [s], the RMSE increased to ~ 0.37 [$^{\circ}$].

In [11], the authors formulated a state prediction algorithm for an Autonomous Underwater Vehicle (AUV) by leveraging Extreme Learning Machines (ELMs). They conducted tests on models of the pitch (θ), pitch rate ($\dot{\theta}$), heave (Z), and heave velocity (w) of an underwater vehicle. They employed a Nonlinear AutoRegressive Moving Average with eXogenous input (NARMAX) framework, which proved effective in forecasting selected Key Performance Indicators (KPIs) over brief periods. The selected data sample period was 0.56 [s], and the study's findings demonstrated acceptable performance with a time delay of less than 1s.

In [12], the authors presented a Deep Neural Network (DNN) method for the prediction of 6-DoF ship motions under real conditions. This method uses a transformer neural network to learn the relationship between ship motions and environmental conditions. The model was trained on a dataset of AIS data records with a period of 1 [s] and bathymetry data, and it was validated by predicting

the motions of a Ro-Pax Passenger ship between two ports in the Gulf of Finland. The results show that the proposed method can predict the rate of ship motions (surge, sway, heave, roll, pitch, yaw) in real conditions with a Mean Absolute Error of 0.49 [m/s], 0.10 [m/s], 0.001 [m/s], 0.005 [°/s], 0.0005 [°/s], and 0.01 [°/s] respectively. The proposed model is suggested for use in collision avoidance and automatic ship control applications.

Table 1: Summary of related work according to reference, forecasted motions, machine learning algorithms, and resulting accuracy.

Reference	Motions	Algorithm	RMSE
[8]	θ, φ, Z	NAR	0.1, 0.05, 0.05
[9]	φ	ARIMA	0.04
[10]	ψ	NN	0.2
[11]	$\theta, \dot{\theta}, Z, w$	ELM	[-]
[12]	$\theta, \varphi, \phi, X, Y, Z$	DNN	0.0161, 0.005, 0.00055, 0.4959, 0.1173, 0.001

3 PROBLEM DESCRIPTION AND AVAILABLE DATA

In this study, the authors investigate the problem of short-term motions forecasting for vessel roll (φ) and trim (ψ). To this end, the authors leverage real-world operational data gathered over a period of one year for a twin diesel engine commercial vessel. The features in the data set can be grouped into two categories: (i) exogenous data, which describes the weather conditions through a number of climate and metocean features; and, (ii) endogenous data, which describes the on-board behaviors, such as the state of the propulsive system, the current position, and the trajectory of the ship. The vessel motions, that is, the roll (φ) and trim (ψ), are a subset of the endogenous data. The dataset is summarized according to source, category, feature, and unit in Table 2, but there are 49 time series features in total (because some features have more than one data-stream).

Additionally, the data are non-continuous, and characterized by 175 portions of varying lengths. There are approximately 500,000 examples sampled over a period of 3 [s]. In fact, for each of the 175 portions, we can determine the weather conditions in which the vessel operates by comparing two common metrics: (i) the wind speed (e.g., according to the Beauford wind scale [13]) and (ii) the sea swell (e.g., according to the Douglas sea state [14]). This allows us to characterize the operating conditions into a number of classes based

on quantitative metrics. Figure 1 summarizes this approach by showing the average wind speed and average sea swell across the 175 time series portions, which can be categorized into 10 classes of weather conditions. Additionally, qualitative descriptions for the Beauford wind scale and the Douglas sea state are included in the figure legend.

Table 2: Dataset summary according to the source, category, feature, and unit. Note that: there are 49 time series features in total (due to some features having more than one data-stream).

Source	Category	Feature	Unit
Exogenous	Climate	Air Temperature	[°C]
		Relative Air Humidity	[-]
		Relative Wind Speed	[m/s]
		True Wind Speed	[m/s]
		Relative Wind Angle	[°]
	Metocean	Mean Wind Angle	[°]
		Mean Wave Period	[s]
		Mean Swell Period	[s]
		Swell + Wave Height	[m]
		Mean Wave Angle	[°]
Endogenous	Main Engine	Speed	[rpm]
		Fuel Consumption	[m ³ /s]
	Diesel Generators	Fuel Consumption	[m ³ /s]
		Load	[%]
	Thrusters	Power	[kW]
		Absorbed Power	[kW]
	Propeller	Pitch Ratio	[-]
		Speed	[m/s]
		Torque	[Nm]
	Rudder	Angle	[°]
	Speed	Speed Over Water	[m/s]
		Speed Over Ground	[m/s]
	Position	Drift Angle	[°]
		Draft at Bow	[m]
Draft at Stern		[m]	
Motion	Roll Angle	[°]	
	Trim Angle	[°]	

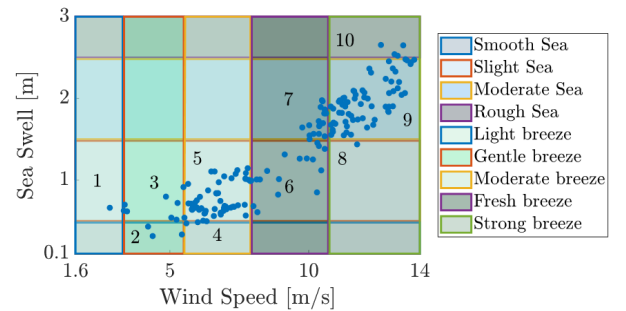


Figure 1: Average wind speed and average sea swell across the 175 time series portions which can be categorized into 10 classes of weather conditions.

4 METHODOLOGY

The approach in this study begins by mapping the problem of vessel motion prediction into a regression framework using Machine Learning. We begin with the conventional framework represented by an input space $\mathcal{X} \subseteq \mathbb{R}^d$, output space $\mathcal{Y} \subseteq \mathbb{R}^b$, and a target phenomenon $\mu : \mathcal{X} \rightarrow \mathcal{Y}$ to be learned [15], [16]. For pointwise motion prediction, \mathcal{X} includes exogenous and endogenous vessel data, excluding motions, whereas \mathcal{Y} pertains solely to the vessel motions (φ and ψ).

Furthermore, short-term motions forecasting requires expanding the regression framework by incorporating two temporal model hyperparameters. First, Δ^- , incorporates historical data, extending the input space to encompass past data from the interval $[t - \Delta^-, t]$ (i.e., $[\mathcal{X}, \mathcal{Y}] \subseteq \mathbb{R}^{d+b}$). The second, Δ^+ , defines the forecast horizon (i.e., the vessel motions at time $t + \Delta^+$). In addition, reliable feature estimations can be used to further enrich the input space within $(t, t + \Delta^+]$.

Attention towards the correct choice of Δ^- is required to balance the dimensionality of the problem with capturing the dynamic effects [15]–[17]. Conversely, the ideal Δ^+ depends on the specific application [15]–[17]. For short-term vessel motions forecasting, Δ^+ should be in the order of a few seconds to ensure an adequate thrust allocation time for the vessel control system.

When selecting a machine learning algorithm for this application, the no-free-lunch theorem [18] requires testing multiple algorithms to find the best one. For the problem at hand, we test three shallow state-of-the-art algorithms from two different families [19], [20]. From the family of Kernel Methods [21], the authors selected to test Kernel Ridge Regression (KRR) using the Gaussian Kernel for the reason described in [22]. While from the family of Ensemble Methods [23], [24] the authors selected to test Random Forests (RF) and XGBoost (XGB) [25].

KRR requires tuning both the regularisation hyperparameter C and the kernel coefficient γ .

RF requires tuning the number of features to be 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 the RF improves with the number of trees n_t , we fixed it to 1000 to keep it computationally tractable.

XGB requires 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

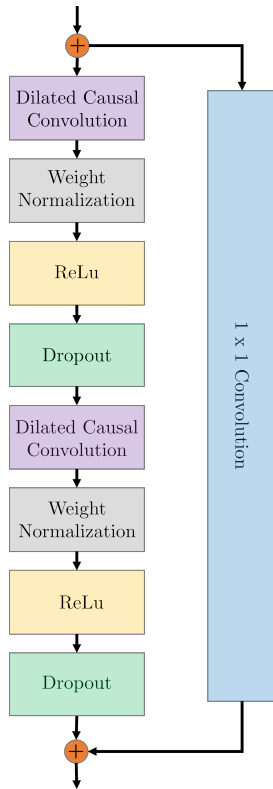
each tree creation n_b , and the number of features to randomly sample from the entire set of features at each node n_f .

Additionally, in line with the current state-of-the-art approaches [26], the authors selected to test a deep learning approach: Temporal Convolutional Network (TCN) [26], [27]. While other deep learning architectures (e.g., the classical and Bidirectional Long Short-Term Memory network [28]) are also suitable candidate architectures for the problem at hand, previous studies have shown that the TCN generally outperforms the other deep learning models while also addressing several of their weaknesses [26], [27]. The TCN architecture illustrated in Figure 2 shows the proposed deep learning model architecture based on the TCN. The general architecture outlined in Figure 2(a), shows the 8 layer TCN block. The first TCN block serves as the input for the original time series signals and, as shown in Figure 2(b), the output of the network was the targets (i.e., the vessel motions) at the desired forecast horizons. For TCN, there are a number of hyperparameters to consider: 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 $k_{s,i}$. For each algorithm, a summary of the hyperparameters with the associated search space is reported in Table 3.

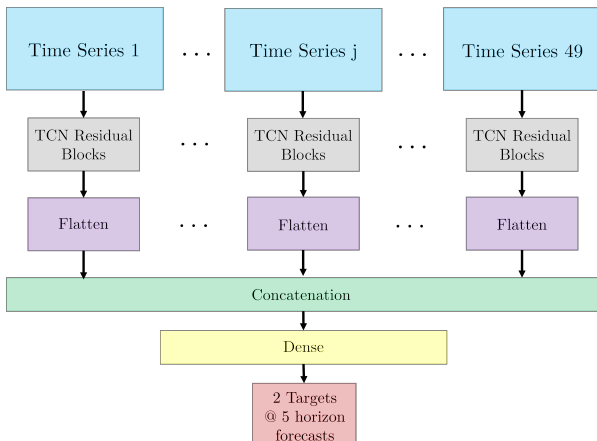
Regarding the implementation of the algorithms for KRR and RF, the models were developed using an in-house custom Python toolbox, for XGB the models relied on the implementation found at [29].

Table 3: Hyperparameters and associated hyperparameter space for each algorithm tested in this work.

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



(a) TCN block layers.



(b) Model architecture using the TCN.

Figure 2: TCN block and model architecture.

To implementat the TCN, the models were developed using custom software relying on the TensorFlow [30] Python module.

To tune the models' hyperparameters and assess the performance of the algorithms, the following Model Selection (MS) and Error Estimation (EE) procedures were employed [17].

For EE, based on the fact that the desired model should be able to extrapolate over unseen weather conditions, the data were divided into Training \mathcal{D}_n and Test \mathcal{T}_t sets using the Leave One Out (LOO) principle applied to different classes of weather conditions (see Section 3). For example, all the data cor-

responding to a single class of weather conditions were allocated into \mathcal{T}_t while the remaining ones are kept in the \mathcal{D}_n .

It is then possible to use \mathcal{D}_n to train the model and select the associated best hyperparameters, and use \mathcal{T}_t to assess the performance of the final model. Repeating this procedure multiple times gives us the average performance in different scenarios (i.e., LOO). Furthermore, because the complexity of this problem increases with the adversity of the weather conditions, we reserved the most complex scenario (class 10) from the learning procedure for the final performance testing to present an unbiased and realistic test of the proposed approach.

Instead, for the MS, namely tuning the hyperparameters of the different algorithms, the following procedure was applied. First, \mathcal{D}_n was split into Learning \mathcal{L}_l and Validation \mathcal{V}_v sets using the same LOO principle as previously described for the EE. Then, for all of the possible hyperparameter configurations (see Table 3), a model was trained on \mathcal{L}_l and its performance was assessed on \mathcal{V}_v according to the Mean Absolute Error (MAE). This procedure was then repeated for each LOO scenario, and the chosen hyperparameter configuration is the one with the lowest MAE when the performance was averaged across all the validation sets. Finally, just before the EE, the model is retrained using the entire \mathcal{D}_n and the best hyperparameter configuration.

This approach is summarized in Figure 3.

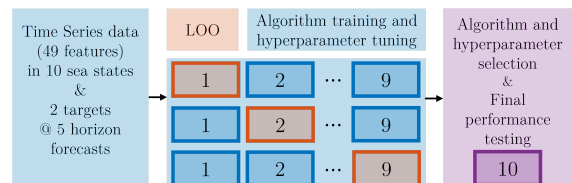


Figure 3: Leave One Out (LOO) methodology for algorithm and hyperparameter selection (on weather classes 1–9) and final performance testing (on weather class 10).

5 RESULTS

This section presents the results obtained by following the methodology proposed in Section 4 using the data described in Section 3.

For the first part, the LOO resampling procedure was carried out with weather condition classes 1 – 9 to determine how each algorithm (KRR, RF, XGB, TCN) performs when forecasting short-term motions while extrapolating over weather conditions. The experiments considered $\Delta^- \in \{3, 12, 48, 64, 128\} [s]$ and $\Delta^+ \in \{3, 6, 12, 24, 48\}$

[s]. The results of this experiment are presented in Tables 4 and 5 for the Trim (ψ) and Roll (φ) motions, respectively. The tables report the MAE for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each of the algorithms, along with the interval of confidence evaluated according to the t-student’s distribution with 95% confidence and $n - 1$ degrees of freedom (where $n = 9$ because of the number of classes in the LOO scenario). Note that a factor of 1×10^{-2} was removed from the results to ensure readability.

Table 4: Trim motion (ψ): MAE [$\times 10^{-2\circ}$] in LOO conditions for weather condition classes 1 – 9 for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each of the algorithms (KRR, RF, XGB, TCN).

Algorithm	Δ^+ [s]					
	3	6	12	24	48	
Shallow	KRR	2.09 ± 0.30	2.51 ± 0.42	2.53 ± 0.55	2.54 ± 0.57	2.54 ± 0.53
	RF	2.10 ± 0.41	2.49 ± 0.45	2.49 ± 0.52	2.54 ± 0.59	2.56 ± 0.56
	XGB	2.07 ± 0.38	2.48 ± 0.41	2.49 ± 0.51	2.49 ± 0.54	2.54 ± 0.55
Deep	TCN	2.01 ± 0.29	2.50 ± 0.44	2.52 ± 0.50	2.53 ± 0.55	2.53 ± 0.58

Table 5: Roll motion (φ): MAE [$\times 10^{-2\circ}$] in LOO conditions for weather condition classes 1 – 9 for different forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for each of the algorithms (KRR, RF, XGB, TCN).

Algorithm	Δ^+ [s]					
	3	6	12	24	48	
Shallow	KRR	1.32 ± 0.12	1.44 ± 0.15	1.47 ± 0.16	1.73 ± 0.22	1.90 ± 0.28
	RF	1.41 ± 0.14	1.47 ± 0.16	1.51 ± 0.18	1.75 ± 0.25	2.02 ± 0.33
	XGB	1.33 ± 0.11	1.43 ± 0.14	1.45 ± 0.15	1.72 ± 0.20	1.98 ± 0.27
Deep	TCN	1.22 ± 0.09	1.31 ± 0.11	1.41 ± 0.13	1.63 ± 0.18	1.91 ± 0.24

Figures 4 and 5 show the varying MAE versus Δ^+ for each of the possible Δ^- combinations in the LOO scenario applied to the weather classes 1–9. From the results in Tables 4 and 5 and Figures 4 and 5, the best algorithm is defined as the one with the lowest MAE at the Δ^+ which is the furthest in the future, but still exhibits a low error. There are a few observations to make. First, as the forecast horizon increases, the error increases; however, for the trim motion, the error saturates at a forecast horizon of up to 6 [s] which is the point where the predictions are no longer reliable (i.e., $\text{MAE}(\psi) \approx 0.025[\circ]$ using the TCN). Second, in general, as the forecast horizon increases the amount of past information included in the prediction (captured in the window $[t - \Delta^-, t]$) should be increased; al-

though, as seen for the roll motion this saturates at approximately $\Delta^- = 64$ [s]. Finally, it is possible to infer that reasonable forecasts can be obtained for the trim motion (ψ) within a Δ^+ of 6 [s]; whereas for the roll motion (φ) within a forecast horizon (Δ^+) of up to 12 [s] (i.e., $\text{MAE}(\varphi) \approx 0.014[\circ]$ using the TCN).

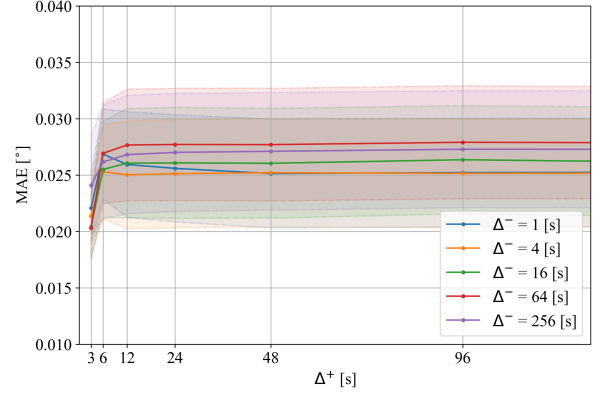


Figure 4: Trim motion (ψ): MAE versus Δ^+ for each of the Δ^- options for the best performing algorithm.

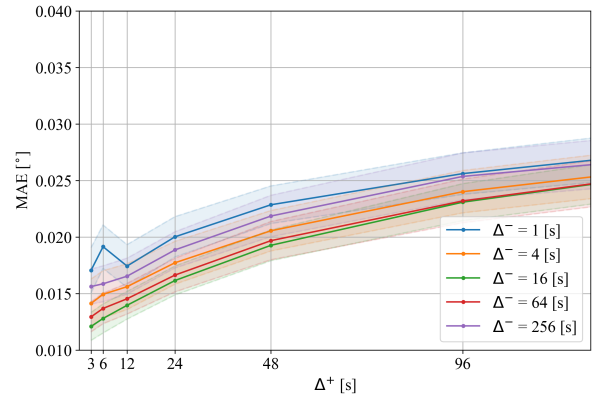


Figure 5: Roll motion (φ): MAE versus Δ^+ for each of the Δ^- options for the best performing algorithm.

Finally, according to the methodology outlined in Section 4, to obtain a more accurate representation of a real world test, the best models will be applied to unseen data coming from the most challenging scenario (weather class 10).

This experiment, aimed at providing an unbiased assessment of the proposed models in a real test scenario, was performed by selecting the best model and temporal hyperparameters for each motion according to Tables 4 and 5 and Figures 4 and 5 for the best possible Δ^+ (i.e., the largest forecast horizon that is still characterized by a low error).

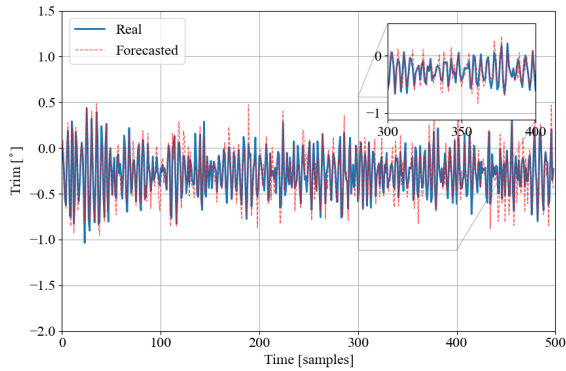


Figure 6: Trim motion (ψ): Real versus Forecasted with $\Delta^+ = 6$ [s] and $\Delta^- = 48$ [s].

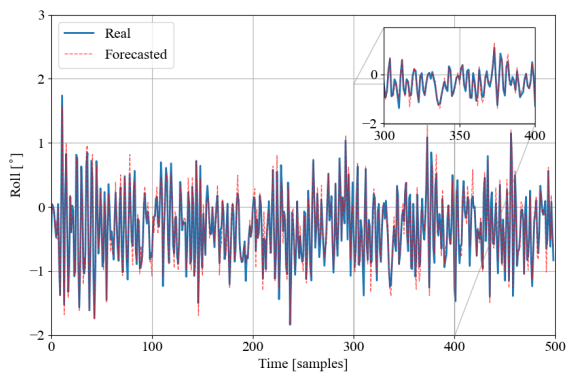


Figure 7: Roll motion (φ): Real versus Forecasted with $\Delta^+ = 12$ [s] and $\Delta^- = 48$ [s].

Table 6: Trim (ψ) and Roll (φ) motions: Best forecast horizons (Δ^+) with the optimal model and temporal (Δ^-) hyperparameters for the best algorithm on weather class 10.

Motion	Algorithm	Δ^+ [s]	Δ^- [s]	MAE [$^\circ$]
ψ	TCN	6	48	0.0251
φ	TCN	12	48	0.0145

Using Tables 4 and 5 and Figures 4 and 5, the horizons are defined as 6 [s] for the roll and 12 [s] for the trim because these forecasts correspond to an error of less than 10% and 5% respectively, which is an acceptable margin for the task at hand.

The results obtained by applying the proposed method to the data left in weather class 10 are shown in Figures 6 and 7, where the real versus forecasted motions are presented for the trim and roll, respectively. Note that, for readability only 500 samples have been reported in the Figures; however, there were approximately 2050 data samples corresponding to weather class 10.

Finally, for a quantitative description of the mod-

els' performance, the error metrics are reported in Table 6 (over the entire portion of data belonging to this class).

Figures 6 and 7, and Table 6 demonstrate the efficacy of the proposed TCN-based short-term motions forecasting. Importantly, and distinct from the other approaches in the literature, the authors demonstrate the TCN-based approach is robust to changes in operating conditions and still performs well during periods of high exogenous loading.

6 CONCLUSIONS

In the fast-paced and demanding landscape of maritime technology, where accurate and precise state prediction is of utmost importance to enable intelligent vessel control systems, the proposed models harness the capabilities of both shallow and deep learning algorithms to deliver high accuracy in short-term vessel motion forecasting. The authors showed that the proposed models are robust, maintaining their predictive accuracy even under challenging conditions characterized by high exogenous loading, which is an important step toward developing fast and reliable short-term motions forecasting algorithms. The demonstrated forecasting framework has been subjected to empirical validation using a dataset collected from an operational vessel over a year.

The results of this study are promising. They show that for trim prediction, our models achieve a forecast horizon of up to 6 seconds, accompanied by a mean absolute error of $2.51[\times 10^{-2}^\circ]$, which translates to a mean absolute percentage error of 9.12%. However, for the roll prediction, the performance is even more impressive, achieving a 12 seconds forecast horizon with a mean absolute error of $1.45[\times 10^{-2}^\circ]$ and a corresponding mean absolute percentage error of 4.64%.

All models, whether based on shallow or deep learning algorithms, exhibited comparable levels of accuracy. There is a noticeable but acceptable decline in accuracy when the prediction horizons are extended to 24 and 48 seconds. For the roll motion, the mean absolute percentage errors at the extended horizons are 9.19% and 12.63%, respectively. Importantly, these errors remain within acceptable margins for making operational-related decisions.

Given the increasing data stream availability at higher sampling rates, the potential for extending these predictive horizons is increasingly likely. While the models developed in this study

are promising, they are validated using data from a single vessel, which poses questions about their generalizability across different types of vessels and operational conditions. Future research should address this by further validating the effectiveness of the deep learning-based approach using more diverse datasets and increased sampling rates for the specific problem at hand.

The next step in this research is the seamless integration of these advanced models into the existing vessel control systems. The successful integration of advanced predictive models into existing vessel control systems offers a unique opportunity to enhance the real-time decision-making processes onboard. In traditional vessel control systems, operators often rely on heuristic methods and past experience to make navigational and operational decisions. The introduction of the proposed models can transform this paradigm by providing data-driven insights that are both fast and accurate. This is particularly crucial in challenging maritime conditions where swift decision making can differentiate between safe navigation and operational hazards.

Moreover, the ability of the proposed models to maintain high levels of accuracy even under conditions of high exogenous loading adds an extra layer of reliability and robustness to the control systems. This is invaluable in scenarios such as heavy weather sailing or navigating through congested waterways, where the margin for error is minimal. Additionally, the models' scalability to longer prediction horizons, while maintaining acceptable error margins, indicates their potential for future applications that require long-term planning, such as route optimization and fuel efficiency calculations.

ACKNOWLEDGMENTS

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. Additionally, the computational portion of this work leveraged the Delft-Blue Supercomputer (Phase 1) [31].

REFERENCES

[1] S. Li and K. S. Fung, "Maritime autonomous surface ships (mass): Implementation and legal issues," *Maritime Business Review*, vol. 4, pp. 330–339, 4 Nov. 2019.

[2] E. Jokioinen, J. Poikonen, M. Hyvönen, *et al.*, "Remote and Autonomous Ships - The next step," *AAWA: Advanced Autonomous Waterborne Applications*, pp. 1–87, 2016.

[3] S. Thombre, Z. Zhao, H. Ramm-Schmidt, *et al.*, "Sensors and ai techniques for situational awareness in autonomous ships: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 64–83, 2022.

[4] International Maritime Organization, *Maritime Safety Committee (MSC), 100th session, 3-7 December 2018*, <https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-100th-session.aspx>, Last visited 11th March 2022, 2018.

[5] A. C. Qi, Q. L. Han, and Y. L. Wang, "A survey of motion control for marine vehicles," *Chinese Control Conference, CCC*, vol. 2015-September, pp. 4214–4218, 2015.

[6] R. Skulstad, G. Li, T. I. Fossen, T. Wang, and H. Zhang, "A Co-Operative hybrid model for ship motion prediction," *Modeling, Identification and Control*, vol. 42, no. 1, pp. 17–26, 2021.

[7] T. Fossen and Ø. Smogeli, "Nonlinear time-domain strip theory formulation for low-speed manoeuvring and station-keeping," *MIC Journal of Modeling, Identification and Control*, vol. 25, pp. 201–221, 2004.

[8] G. De Masi, R. Bruschi, and F. Gaggiotti, "Short term vessel motion forecasting based on wavelet neural network for wave feed-forward dynamic positioning," in *Proceedings of the International Offshore and Polar Engineering Conference*, 2012, pp. 915–919.

[9] N. Suhermi, Suhartono, D. D. Prastyo, and B. Ali, "Roll motion prediction using a hybrid deep learning and ARIMA model," *Procedia Computer Science*, vol. 144, pp. 251–258, 2018.

[10] W. Zhang and Z. Liu, "Real-time ship motion prediction based on time delay wavelet neural network," *Journal of Applied Mathematics*, vol. 2014, 2014.

[11] B. N. Rath and B. Subudhi, "Chapter 12 - an extreme learning-based adaptive control design for an autonomous underwater vehicle," in *Cognitive Informatics, Computer Modelling, and Cognitive Science*, G. Sinha and J. S. Suri, Eds., 2020, pp. 213–228.

[12] M. Zhang, G. Taimuri, J. Zhang, and S. Hirdaris, "A deep learning method for the prediction of 6-dof ship motions in real conditions," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 0, no. 0, p. 14 750 902 231 157 852, 0.

[13] National Weather Service, *Beaufort Wind Scale*, <https://www.weather.gov/mfl/beaufort>, Accessed August 2023, 2023.

[14] World Meteorological Organization, *Marine Frequently Asked Questions: 3. What is the Douglas scale and is it appropriate terminology?* https://www.wmo.int/pages/prog/education/faq_marine/

- // community.wmo.int/en/activity-areas/Marine/About/FAQs, Accessed August 2023, 2023.
- [15] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [16] J. D. Hamilton, *Time series analysis*. Princeton university press, 1994.
- [17] L. Oneto, *Model Selection and Error Estimation in a Nutshell*. Springer, 2020.
- [18] S. P. Adam, S. A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, “No free lunch theorem: A review,” *Approximation and optimization*, pp. 57–82, 2019.
- [19] 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*, vol. 15, no. 1, pp. 3133–3181, 2014.
- [20] M. Wainberg, B. Alipanahi, and B. J. Frey, “Are random forests truly the best classifiers?” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3837–3841, 2016.
- [21] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [22] S. S. Keerthi and C. J. Lin, “Asymptotic behaviors of support vector machines with gaussian kernel,” *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.
- [23] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [24] I. Orlandi, L. Oneto, and D. Anguita, “Random forests model selection,” in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.
- [25] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *ASM SIGKDD international conference on knowledge discovery and data mining*, 2016.
- [26] 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)*, IEEE, 2022, pp. 01–10.
- [27] V. D’Amato, L. Oneto, A. Camurri, et al., “The importance of multiple temporal scales in motion recognition: From shallow to deep multi-scale models,” in *IEEE International Joint Conference on Neural Networks*, 2022.
- [28] L. Zhu, X. Wang, and J. Gao, “Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks,” *arXiv preprint arXiv:1603.07772*, 2016.
- [29] 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*, San Francisco, California, USA: ACM, 2016, pp. 785–794.
- [30] M. Abadi, A. Agarwal, P. Barham, et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, <https://www.tensorflow.org/>, 2015.
- [31] Delft High Performance Computing Centre (DHPC), *DelftBlue Supercomputer (Phase 1)*, <https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1>, 2022.