Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Modeling Vessel Shaft Power from Noon Reports: Data Fusion Strategies for Deep Learning under varying Data Availability

Akriti Sharma

akriti@simula.no

Simula Research Laboratory AS https://orcid.org/0000-0003-4623-7938

Dogan Altan

Simula Research Laboratory

Joachim Haga

Simula Research Laboratory

Glenn Lines

Simula Research Laboratory

Dusica Marijan

Simula Research Laboratory

Arnbjørn Maressa

Navtor (Norway)

Research Article

Keywords: Multi-source data fusion, Vessel performance monitoring, Cross-fleet transferability, Cross-temporal generalization, Data sufficiency

Posted Date: August 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10788923/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Modeling Vessel Shaft Power from Noon Reports: Data Fusion Strategies for Deep Learning under varying Data Availability

Akriti Sharma[1*], Dogan Altan[1], Joachim Haga[1], Glenn Lines[1], Dusica Marijan[1], Arnbjørn Maressa[2]

1Simula Research Laboratory, Kristian Augusts Gate 23, Oslo, 0164, Norway.

2Navtor AS, Elganeveien 1, Egersund, 4373, Norway.

*Corresponding author(s). E-mail(s): akriti@simula.no;

Abstract

Accurate shaft power prediction under dynamic ocean conditions is important for optimizing maritime operations and estimating fuel consumption, yet remains challenging when low-resolution noon reports are the only available source of vessel performance data. Existing approaches predominantly rely on a single data source at a fixed temporal resolution, limiting their generalization. This paper addresses this gap by proposing three deep learning approaches that predict shaft power by fusing noon reports with external data sources that differ in temporal resolution, spatial coverage and reliability. The proposed approaches integrate these sources through fusion strategies tailored to different data availability scenarios, ranging from vessels with extensive operational histories to those with no historical data. The approaches are evaluated on two heterogeneous fleets comprising bulk carriers and oil tankers under different preprocessing conditions and temporal resolutions. Results demonstrate that the proposed fusion strategies substantially improve predictive accuracy over sparse noon report baselines and enable models trained on low-frequency noon reports to generalize to high-frequency power predictions, narrowing the resolution gap between the two data sources. Cross-fleet transferability across heterogeneous vessel categories is established, minimum training data requirements for reliable deployment are identified, and practical guidance for selecting the appropriate approach is provided.

Keywords: Multi-source data fusion, Vessel performance monitoring, Cross-fleet transferability, Cross-temporal generalization, Data sufficiency

1 Introduction

Maritime transport forms the foundation of global trade, accounting for around 80% of the total volume of international trade [1]. This makes optimizing maritime operations essential for reducing operational costs and greenhouse gas emissions. In particular, improving vessel operational efficiency through measures such as speed optimization and accurate shaft power prediction directly influences fuel consumption and, consequently, emissions [2].

Vessel performance monitoring involves processing data from several heterogeneous sources. Onboard sensors record operational variables at high frequency to provide detailed insight into vessel behavior. Noon reports (NRs) are daily logs maintained by the vessel crew at a fixed time each day. These reports record daily averaged navigational, operational, and environmental data, reflecting a vessel's sailing profile over the reporting period. Although sensor data offer higher accuracy, their availability remains limited due to substantial installation and maintenance costs, and many vessels operate without continuous sensor coverage. On the other hand, NRs are widely available across fleets and vessel types, making them a practically accessible data source.

Several external data sources can support vessel performance monitoring, including Automatic Identification System (AIS) messages that provide voyage-related information [3], publicly available meteorological data on wind and waves from ERA5, and sea current data from the Copernicus Marine Environment Monitoring Service (CMEMS) [4]. Weather and sea-state conditions are critical for maritime operations due to their influence on navigation safety and vessel performance [5], though the reliability of

these sources is not always guaranteed, as AIS data are known to suffer from various data quality issues [6, 7]. Effective integration of heterogeneous data sources is essential for accurately predicting vessel behavior [8], particularly shaft power, due to its close relationship with fuel consumption. Despite growing interest in fused datasets for maritime prediction, it remains unclear which modeling approaches are best suited to varying data availability scenarios, posing challenges for practical deployment.

This paper presents three deep learning approaches for shaft power prediction from NRs, each integrating data from additional sources through a distinct fusion strategy tailored to a different data availability scenario. Rather than considering the sparsity of NRs as a limitation, the first approach leverages their structure under a weakly supervised group-level objective. The second transfers knowledge from sensor data of a source vessel and fine-tunes using the target vessel NRs. The third learns temporal patterns embedded in NRs by reconstructing sensor sequences as an auxiliary task. The approaches are based on well-established machine learning methods: weakly supervised learning, transfer learning, and reconstruction-guided learning. The contribution of this study is adapting these methods to the limitations of NR data and combining them into a single framework for vessels with different levels of data availability. The approaches are systematically evaluated across two heterogeneous fleets, varying preprocessing conditions, and multiple temporal resolutions to assess their suitability across different settings. This evaluation is guided by four research questions (RQ) (detailed in Section 6.1), addressing data sparsity, cross-temporal generalization, training data sufficiency and cross-fleet transferability, respectively. The main contributions of the paper can be summarized as follows:

- We present three approaches for shaft power prediction from low-frequency NRs, each integrating additional data sources through a distinct fusion strategy (RQ1).
- We demonstrate generalization from NRs to high-resolution power prediction (RQ2).
- We establish minimum training data requirements for reliable predictions under each approach (RQ3).
- We establish cross-fleet transferability across heterogeneous vessel types and provide practical recommendations for approach selection (RQ4).

This paper is organized as follows. First, we review the literature on vessel shaft power prediction, followed by a description of the dataset and relevant background. Next, we present the proposed approaches and their experimental evaluation. Finally, we discuss the findings and outline future research directions.

2 Related work

Methods for performance prediction, including shaft power prediction and fuel oil consumption prediction, have been intensively investigated in the literature [2, 9]. In this section, we summarize the related work on the shaft power prediction problem by presenting the earlier studies under three headings: (a) data-driven approaches (b) objective/loss adaptation-based approaches and (c) transfer learning-based approaches.

2.1 Data-driven approaches

Kiouvrekis et al. [10] analyze machine learning models including k-NN, random forests and neural networks, along with SHapley Additive exPlanations (SHAP) [11] to quantify the contribution of input features in shaft power prediction. Similarly, Kim et al. [12] investigate random forest regressors with SHAP for prediction and feature contribution analysis.

Laurie et al. [13] employ five machine learning models for propulsion power prediction and additionally investigate fouling-related performance degradation. Similarly, Vorkapić et al. [14] analyze linear regression, multilayer perceptrons, support vector machines and random forests for fuel consumption prediction, whereas [15] shows that random forests outperforms for propulsion power prediction.

Furthermore, Kim et al. [16] investigate support vector regression using onboard sensors and ocean data, Uyanık et al. [17] examine data-driven methods for shaft power and fuel consumption prediction and Liang et al. [18] study traditional and deep learning approaches to predict vessel propulsion power using AIS, weather and sensor data. Parkes et al. [19] investigate the use of wave height as a feature for vessel power prediction using artificial neural networks. Parkes et al. [20] analyze the extrapolation capability of neural networks using fused multi-vessel data, while Karagiannidis and Themelis [21] study neural networks and model parameter choices for predicting vessel fuel consumption.

Such approaches learn input-output mappings from vessel-specific datasets (either fused or not) without explicitly modeling the influence of operational and environmental conditions across vessel configurations, which limits their generalization potential. This study addresses that gap by systematically constructing and evaluating fused multi-source datasets for shaft power prediction under low-frequency data regimes.

2.2 Objective adaptation-based approaches

Vessel dynamics represented through physical constraints have also been investigated. Studies demonstrate the effectiveness of incorporating such constraints into learning objectives [22–24]. Parkes et al. [24] have examined neural networks of varying sizes to model the relationship between vessel operating conditions and shaft power. Similarly, Bourchas and Papalambrou [25] introduce a physics-informed neural network approach for estimating a vessel engine's main power by embedding the propeller law into the loss function alongside the conventional mean squared error. Altan et al. [23] propose a neural network architecture that combines multiple data sources while incorporating empirically derived physical constraints into the learning objective for shaft power prediction. Mohammed et al. [22] employ Kolmogorov–Arnold Networks [26] to predict shaft power and fuel consumption using empirical physical constraints. Overall, these studies utilize fused datasets and physical constraints to improve shaft power and fuel consumption prediction

2.3 Transfer learning-based approaches

This subsection summarizes prior work employing transfer learning (TL) for shaft power prediction and related maritime tasks such as fuel consumption and vessel speed prediction. TL has shown potential in several maritime applications, including fuel consumption estimation [27]. Mavroudis and Tinga [28] propose a two-stage approach in which a surrogate model is first trained on synthetic physics-based data and later fine-tuned using real operational data. Similarly, Fan et al. [29] apply TL with bidirectional LSTMs and random forests for fuel consumption prediction in scenarios where target vessels lacked sufficient historical data, enabling knowledge transfer from similar vessels. During the TL phase, the initial layers are frozen, and the last layers are fine-tuned on the target vessel data. Feature transfer has also been studied to address differences in data distributions across maritime datasets, with Guo et al. [30] proposing a domain-adapted transfer strategy for short-term vessel speed prediction. TL has further been applied in related maritime contexts, including composite pressure vessel behavior prediction [31] and bearing displacement estimation in multi-support shafting systems [32].

A shared limitation across these studies is their reliance on a single data source and temporal resolution. As a result, knowledge transfer across data collected at different sampling frequencies has received little attention. Most approaches have focused on sister vessels or fleet-wide adoption, and transferability across heterogeneous vessel categories remains unexplored. Their practical applicability is further limited by the use of random data splits rather than temporal validation, which more closely represents real-world forecasting scenarios.

Our previous work [33] addresses some of these gaps by proposing a cross-frequency transfer learning (TL) framework, in which a model trained on a vessel's sensor data is fine-tuned using noon reports (NRs) from sister, similar, and structurally dissimilar vessels within a single fleet. However, Sharma et al. [33]

incorporate weather information only into the sensor data, rather than enriching the NRs directly. Moreover, the study focuses exclusively on evaluating the transfer learning framework, without investigating training-data sufficiency or transferability across heterogeneous fleets. This study extends that work in three key ways. First, methodologically, it integrates AIS, CMEMS, and ERA5 data directly into NRs, enabling two additional learning strategies for scenarios in which no suitable source vessel is available, as envisioned in the future work of Sharma et al. [33]. Second, experimentally, it evaluates these approaches across two heterogeneous fleets rather than vessels within a single fleet. Third, in terms of scope, it quantifies the minimum amount of training data required to achieve reliable prediction both within and across fleets, an aspect that has not been investigated previously.

3 Data

We use datasets from two heterogeneous fleets, A and B. Fleet A has eight bulk carrier vessels with lengths in the range of 204-208 meters, a breadth of approximately 32.3 meters, and the main engine with maximum continuous ratings (MCR) ranging from 10,780 kW to 11,900 kW. Fleet B has seven oil product tankers with lengths of 274-336 meters, breadths of 48-60 meters, and MCR ranging from 13,490 kW to 24,510 kW. The data is provided by our industrial partner. To preserve anonymity, vessels are identified using the fleet's reference followed by a numeric code such as A-1 and B-1.

Sensor data are recorded at a temporal resolution of 15 minutes and referred to as high-frequency data. NRs are present at a lower temporal resolution, typically every 24 hours. NRs are referred to as low-frequency data. Within each fleet, vessels contain data spanning multiple years. Fleet A includes both sensor data and NRs for vessels from 2020 to 2025. Fleet B includes only NR data, available from 1999 to 2024. However, Table 1 provides a detailed description of the training and testing periods for vessels in both fleet A and B. The first column identifies the vessel, followed by the corresponding training and testing data ranges. The selected periods ensure that each dataset contains at least 350 instances from the NRs. The same training and testing periods are also applied to the sensor data. Although high-frequency data will have more than 350 observations, the duration of the periods is kept consistent to ensure a fair comparison across data sources.

Table 1 Training and testing periods for vessels in fleet A and fleet B.

V	Training period	Testing period
A-1	2021-11-01→2023-08-31	2023-09-01→2025-12-31
A-2	2021-08-01→2023-10-31	2023-11-01→2025-12-31
A-3	2022-05-01→2023-11-30	2023-12-01→2025-12-31
A-4	2021-10-01→2023-11-30	2023-12-01→2025-12-31
A-5	2022-01-01→2023-10-31	2023-11-01→2025-12-31
A-6	2021-02-01→2022-09-30	2022-10-01→2025-12-31
A-7	2022-05-01→2024-01-31	2024-02-01→2025-12-31
A-8	2022-01-01→2023-06-30	2023-07-01→2025-12-31
B-1	2023-01-01→2023-12-31	2024-01-01→2024-12-31
B-2	2020-01-01→2022-05-31	2022-06-01→2024-12-31
B-3	2021-01-01→2022-12-31	2023-01-01→2024-12-31
B-4	2020-01-01→2022-12-31	2023-01-01→2024-12-31
B-5	2020-01-01→2022-12-31	2023-01-01→2024-12-31

V	Training period	Testing period
B-6	2020-01-01→2022-06-30	2022-07-01→2024-12-31
B-7	2020-01-01→2022-06-30	2022-07-01→2024-12-31

3.1 Data fusion using multiple sources

In this study, vessel data are used to predict shaft power as the outcome variable. The dataset consists of NRs and sensor data combined with ship operational and weather information derived from AIS messages, CMEMS and ERA5 data. These additional data sources are used to upsample NRs to match the high temporal resolution of sensor data. To address incorrect or missing weather-related measurements in sensor recordings, values from these sources are used as replacements. Although these sources provide information on vessel operations and encountered weather and sea-state conditions, they do not contain shaft power measurements, resulting in the absence of the outcome variable.

AIS provides vessel navigation and positional information, including timestamp, vessel position (latitude and longitude), speed over ground (SOG) and course over ground (COG). In contrast, CMEMS and ERA5 provide high-temporal-resolution oceanographic and meteorological data, including historical information on wind and wave conditions encountered by the vessels. SOG and COG from AIS messages are combined with ocean current velocity (speed CS and direction CD) from CMEMS to derive vessel operational data such as speed through water (Vstw). Vessel and ocean current velocity are first transformed into Cartesian components:

$$Vx = SOG \cdot sin(COG),\ \ Vy = SOG \cdot cos(COG) \tag{1}$$

$$Cx = CS \cdot sin(CD),\ \ Cy = CS \cdot cos(CD) \tag{2}$$

Vessel velocity is then obtained by subtracting the current velocity components:

$$Lx = Vx - Cx,\ \ Ly = Vy - Cy \tag{3}$$

Finally, Vstw is calculated as the magnitude of the relative velocity vector:

$$Vstw = \sqrt{(Lx^2 + Ly^2)} \tag{4}$$

where (Vx) and (Vy) represent vessel velocity components, (Cx) and (Cy) represent ocean current velocity components, and (Lx) and (Ly) represent the vessel velocity relative to water. Wind, wave, and ocean current information obtained from meteorological sources is present in absolute directions. Therefore, the actual course obtained from vessel data is used in transforming these values to ship-relative directions, where (0°) indicates approaching from the front and (180°) indicates approaching from the aft.

For fusion after data derivation, NRs are parsed and expanded into 15-minute interval entries between their start and end timestamps. AIS data corresponding to each timestamp are fused using linear interpolation, after which data from CMEMS and ERA5 are retrieved based on the AIS-derived vessel location and fused through trilinear interpolation. In the subsequent text, NRs integrated with additional data are referred to as 'fused NRs', whereas the original NRs are referred to as 'non-fused NRs'. In the fused reports, original NR entries are preserved, and only the missing intermediate timestamps at 15-minute intervals are populated from external sources.

3.2 Feature distillation

Vessel data contains more than 50 features describing vessel operation, weather, and sea state in both fleet A and B. However, we select only five features as input for predicting shaft power. The input features are selected based on their physical relationship with hull resistance and propulsive power demand: speed

through water (Vstw in knots) and draft amidships (Vdraft in meters) affect hydrodynamic resistance, while wave height (hwave in meters), wave direction (dwave in degrees), and wind direction (dwind in degrees) capture added resistance from weather and sea state [5]. Pearson correlation with the outcome variable is then used to confirm that these features also have significant linear association with shaft power. Shaft power is measured in kilowatts (kW). Feature selection is also guided by the data fusion strategy. Since NRs are fused using publicly available (AIS, CMEMS, and ERA5) data, only variables available from these sources or already present in vessel's non-fused NRs across fleets can be used. For example, propeller shaft speed (RPM), although physically related to shaft power, is unavailable in the vessel NRs and public data sources used in this study. Thus, it is excluded from the feature set.

Here, Vdraft is calculated as the average of fore and aft drafts. hwave represents the significant wave height and dwave and dwind are relative to the vessel. Directional features are further cosine encoded to capture their contribution relative to the vessel's movement in the forward direction, resulting in the features Rel.dwave and Rel.dwind. The Rel.Dwind feature is additionally weighted by the relative wind speed. Subsequently, the original directional variables are removed from the data, while Rel.dwave and Rel.Dwind are retained. The final input features are:

$$X = [Vstw, Vdraft, hwave, Rel.dwave, Rel.dwind]^T \in \mathbb{R}^5$$

Input features X are standardized using StandardScaler, the outcome variable is normalized to the range [0, 1] using MinMaxScaler from Scikit-learn.

4 Background

The proposed approaches are based on three methodological paradigms: weakly supervised learning, TL and sequence-to-sequence reconstruction. We briefly review each below:

4.1 Regression and MLP

Regression is a machine learning approach for modeling the relationship between the outcome variable and the input variables. The approach estimates a function that maps input variables to a continuous output for predicting numerical values. While regression captures linear relationships between variables, many real-world problems involve nonlinear patterns that cannot be represented by a single regression equation. This is where the multi-layer perceptron (MLP) extends the idea further. MLP is an artificial neural network comprising an input layer, one or more fully connected hidden layers, and an output layer, with nonlinear activation functions applied between layers to capture complex relationships. Each layer is fully connected to the next layer, and data flows in one direction from input to output; hence, it is called a feedforward network. An MLP is trained by minimizing a loss function through backpropagation that iteratively adjusts weights across all layers using gradient descent.

4.2 Weakly supervised learning

Weakly supervised regression: Weakly supervised regression is a regression approach used when the outcome labels for all instances are unavailable. Instead of exact labels for each sample, the model receives group-level information such as group means, interval bounds, or noisy labels. The model is optimized through a loss function defined over aggregated predictions rather than individual ones. In MLP regression, the network still predicts for each input, but parameters are updated by a group-level loss function during backpropagation. The model gradually learns reasonable individual predictions despite receiving supervision at the group level. It enables the model to learn overall patterns in groups and preserves the average behavior during training. The approach is practically valuable in domains where collecting exact ground truth is difficult or physically impossible.

Multiple instance learning (MIL): The group-level loss function falls under the scheme of MIL. In MIL, models are trained using groups of samples instead of individual samples. Each group is called a bag, and the samples inside the bag are called instances. Supervision is provided only at the bag level, meaning the model receives a single outcome label per bag and must learn instance-level predictions that are collectively consistent with the bag-level supervision. The loss function is computed using aggregate statistics of the bag. The aggregate statistics can be the mean, maximum, or minimum of the instance-level predictions. During backpropagation, gradients flow from the bag-level loss through the aggregation operation to all instances in the bag. It allows the network to adjust its weights based on collective error and learn reliable instance-level predictions.

Learning from label proportions (LLP): LLP is a type of weakly supervised learning where the model is trained to match the mean of the outcome values of the group. Unlike MIL, where the bag label may represent different aggregate statistics, LLP specifically uses mean-based supervision. The training groups are predefined, and the loss function is computed between the predicted group mean and the corresponding observed group-level outcome. During inference, the model predicts at the instance level, with the prediction distribution being consistent with the group proportions observed during training. LLP is particularly suitable for problems where data exists in predetermined groups, and the available outcome label reflects an aggregate statistic for instances within that group.

4.3 Transfer learning

Transfer learning (TL): TL is a machine learning approach where knowledge learned by a model during training on a source task or dataset is leveraged to improve model's performance on a different but related outcome or dataset. MLP (base model) is first trained on a large-scale benchmark dataset, and the learned representations are embedded in its weights. Fine-tuning adapts this pre-trained MLP to an outcome, where selected layers are retrained using target-specific data.

There exist different fine-tuning strategies:

- Feature extraction: Pre-trained layers are frozen, and only a newly added or output layer is trained. This is effective when the source and the outcome are closely related.
- Partial fine-tuning: Selected deeper layers are unfrozen and retrained with the output layer to adapt high-level features while preserving low-level representations.
- Full fine-tuning: All layers are retrained using the target dataset and used when the target dataset is large and differs from the source domain.

4.4 Two-phase reconstruction-guided learning

Sequence-to-sequence reconstruction (seq2seq): A seq2seq model is designed to learn a mapping from an input sequence to an output sequence, which may differ in length and structure. Originally developed for machine translation, these models are widely used in time-series forecasting and representation learning. It is built upon an encoder-decoder architecture. The encoder compresses the input into a compact latent representation, and the decoder uses this representation to reconstruct the actual sequence autoregressively or by repeating the latent input across time steps. The reconstruction objective trains the model to minimize the difference between the predicted and actual sequences, forcing the encoder to learn informative latent representations. This self-supervised approach requires no external labels, making it suitable for settings with limited or unavailable labeled data.

Two-phase training: Two-phase training is an optimization strategy in which a model is trained in two sequential stages with different objectives and parameter configurations. In the first phase, only a subset of the model parameters is optimized while the remaining parameters are frozen, allowing the active components to develop a well-structured internal representation without interference from other network

parts. In the second phase, the frozen parameters are unfrozen, and the entire model is jointly fine-tuned with a lower learning rate to refine the learned representations under a broader or combined objective. Since, joint training from the beginning can hinder effective learning due to conflicting gradient signals, causing premature overfitting. Therefore, this staged approach can be useful in multi-task settings where one task is intended to shape the representation before a second task is introduced.

5 Methodology

Our research proposes three approaches for modeling shaft power predictions using NRs. NRs are fused with auxiliary data from additional sources. The functional and architectural design of each approach determines how NRs are integrated with data from external sources.

Figure 1 illustrates the high-level framework for integrating the proposed approaches to predict shaft power using NRs. The framework consists of two parts. Part (A) presents the data fusion strategy. Part (B) presents the conceptual design of each approach and its integration within the overall framework. A baseline regression model, referred to as the "Power Predictor", is first introduced, and later we outline how its architecture and data handling are extended under each proposed approach for shaft power predictions.

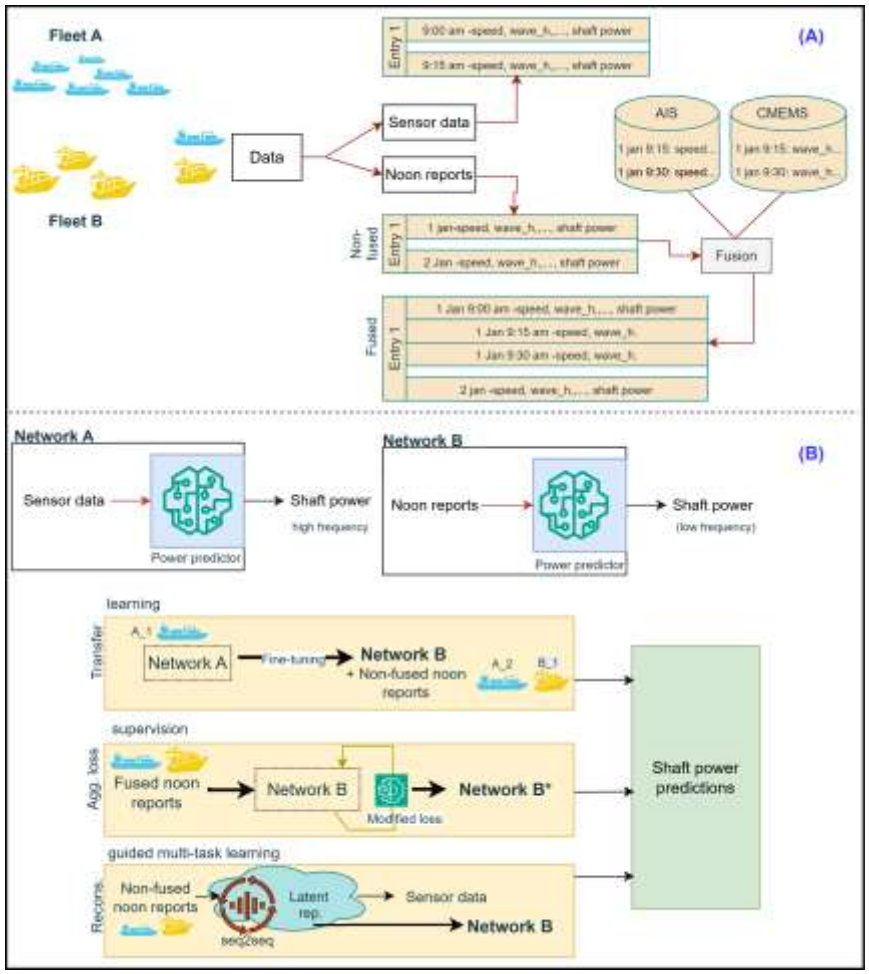


Fig. 1 System Architecture

5.1 Power predictor: base model

Power predictor is a feedforward MLP regression model for modeling the relationship between shaft power (outcome variable) and distilled input feature space (input variables). When the power predictor is trained on sensor data for high-temporal-resolution predictions, it is referred to as network A, whereas the same configuration trained on NRs for low-resolution predictions is referred to as network B. Let $x \in \mathbb{R}^5$ denote the input and $\hat{y} \in \mathbb{R}$ denote the predicted shaft power. The power predictor is a four-layer MLP, which takes a five-dimensional input and passes through three hidden layers with 128, 64 and 32 neurons, respectively, and the output layer with a size of one predicts the shaft power value. The ReLU activation function is used in the hidden layers, while the output layer uses no activation function. The architecture of the power predictor is inspired by a study predicting vessel shaft power using fused data from multiple sources [23].

Given a training set of N individual observations, the model is optimized by minimizing the mean absolute error (MAE) using the Adam optimizer with a learning rate of 10^{-4}.

$$L_base = (1/N)\ \Sigma_{i=1}{}^{N} |f(x_i) - y_i| \quad (5)$$

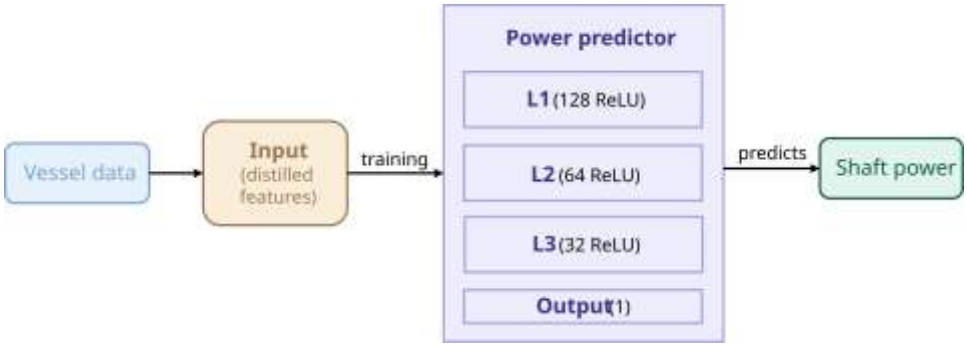


Fig. 2 Power predictor: architectural design and overall pipeline.

This loss is robust to outliers and is suitable for regression tasks with extreme values. This per-instance supervision assumes that a ground truth label y_i is available for each instance and provides instance-level supervision during backpropagation. Training is conducted using 200 epochs with an early stopping strategy implemented with a patience of 10 epochs and a minimum improvement threshold of 1×10^{-4}. During training, a checkpoint mechanism saves the model weights corresponding to the lowest validation loss. Furthermore, an adaptive learning rate schedule was incorporated using the ReduceLROnPlateau strategy, halving the learning rate after three epochs without validation loss improvement. After training, the model restores to the weights with the lowest validation loss across all training epochs. Figure 2 illustrates the block diagram representation of the power predictor architecture and the overall pipeline.

5.2 Power predictions using aggregate loss supervision

The aggregate loss supervision approach, ALS_App, predicts shaft power from fused NRs by introducing a group-aware loss formulation into network B. The approach addresses the absence of instance-level shaft power labels in fused NRs by replacing the per-instance loss of the power predictor with a group-level LLP objective. Fused NRs provide upsampled features at 15-minute intervals, but the shaft power ground truth is only available as a daily aggregate from the original non-fused NRs.

In ALS_App, fused NRs are organized into G groups, where each group g represents a single NR period. Each group contains a variable number of upsampled observations that serve as the instances within the group. Let $Bg = \{(xg,1, \ldots, xg,ng)\}$ denote the g-th bag containing ng instances, where $xg,i \in \mathbb{R}^5$ is the feature vector of the i-th instance. The power predictor defined in Equation (1) is applied to each instance in every bag using a batched forward pass:

$$\hat{y}g,i = f(xg,i),\ i = 1,\ldots,ng,\ g = 1,\ldots,G \quad (6)$$

This produces instance-level power predictions {ŷg,i} for each bag g. Since instance-level ground truth is unavailable, supervision is applied only at the bag level using an LLP objective. The group-level loss is then computed as MAE between the predicted $\hat{y}g$ and observed $\bar{y}g$ group means:

$$Lg = |\hat{yg} - \bar{y}g| \quad (7)$$

The aggregate loss is then defined as MAE between predicted and observed group means, averaged across all G bags:

$$L_ALS = (1/G)\, \Sigma g_{=1}{}^{G} Lg = (1/G)\, \Sigma g_{=1}{}^{G} |\hat{yg} - \bar{y}g| \quad (8)$$

This replaces the per-instance loss in Equation (2). For predictions, the final shaft power is given by averaging the instance-level predictions within each group:

$$\hat{y}g = (1/ng)\, \Sigma_{i=1}{}^{ng} f(xg,i) \quad (9)$$

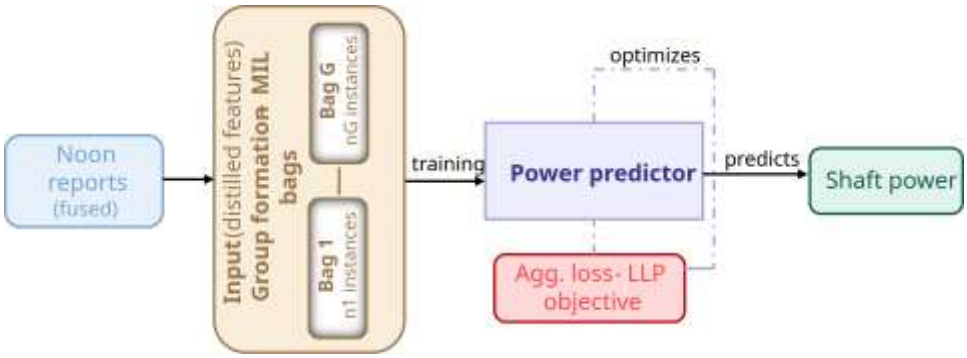


Fig. 3 Power predictions using ALS_App: group-aware LLP objective applied to power predictor over MIL bags of fused NRs.

Figure 3 illustrates the architecture and the overall pipeline for ALS_App in a block diagram format. The fused NRs are preprocessed and partitioned into MIL bags, with one bag per reporting period. The power predictor processes each instance independently, while the aggregate loss computed from group means optimizes the model parameters. The training configuration follows that of the power predictor, with an early stopping patience of five epochs.

The aggregate loss computes the MAE between predicted and actual group means to ensure accurate average predictions. To prevent the model from predicting identical values for each input in a group, we extend the loss calculation with a diversity penalty based on 1/variance, where variance represents the variance of predictions within each group. This should ensure diversity in predictions while maintaining mean accuracy. Moreover, we constrain the prediction spread by additionally penalizing deviations from the outcome standard deviation, trying to guide the model toward a stable level of diversity. However, these extensions led to overfitting on the training data for most vessels. To mitigate this, dropout regularization was introduced after the second and third layers of the power predictor, but it resulted in marginal improvement. Therefore, the basic aggregate loss scheme described above is used finally.

5.3 Power predictions using TL

The TL approach, TL_App, employs a TL-based feature extraction strategy to transfer representations learned from sensor data (network A) to power prediction using NRs (network B). Network A, the power predictor trained on sensor data from a source vessel acts as the pre-trained base model. All layers except the output layer are frozen, and the output layer is retrained using non-fused NRs from target vessels. Unlike ALS_App, TL_App utilizes non-fused NRs under network B. The suitability of a source vessel for TL_App and whether vessel similarity helps in transfer performance are investigated empirically in Section 7.6.

Vessel similarity combines structural and operational similarity. Structural similarity is based on physical features, including vessel type, engine maximum continuous rating (MCR), rated engine speed (RPM), service speed, and vessel dimensions. Operational similarity is characterized by the interquartile ranges of key input features (speed through water, draft amidships, shaft power, wave height, and wind speed) derived from historical voyage data. We devise a quantitative compatibility index for measuring source-target vessel relationships. The index is evaluated using both structural similarity and operational similarity, following the formula:

$$C = (1/n)\,\Sigma_{i=1}^{n} S_i, \quad S_i = 1 - |src_i - tgt_i| / max(src_i, tgt_i) \tag{10}$$

where src_i and tgt_i represent the value of the source vessel and target vessel, respectively, for each feature i. S_i is the similarity score for the i-th feature and ranges between 0 and 1. C is the overall compatibility score, computed with no manually assigned weights across features, and n represents the total number of features.

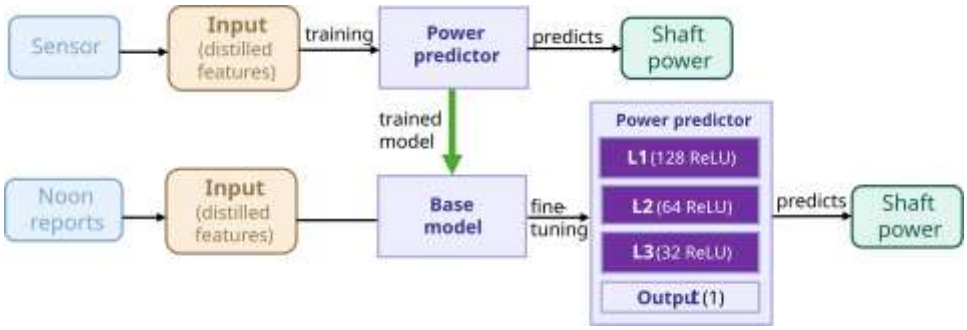


Fig. 4 Power predictions using TL_App: pre-training on source sensor data and fine-tuning on target vessel NRs. Layers with frozen weights are grayed out in the fine-tuned power predictor.

Network A follows the base power predictor training configuration with early stopping patience reduced to five epochs. Once trained, weights and data scaling functions are saved. During fine-tuning, the pre-trained weights from the base model are loaded, all layers except the output layer are frozen, and the scaling functions from network A are transferred directly to the NRs without refitting. The fine-tuning objective and training configuration follow those of the power predictor, with batch size 32. Figure 4 illustrates the complete TL_App pipeline in a block diagram format.

5.4 Power predictions using reconstruction-guided multi-task learning

The reconstruction-guided multi-task learning approach RGML_App predicts shaft power from non-fused NRs by employing seq2seq reconstruction as an auxiliary task. RGML_App requires the model to simultaneously reconstruct sequences of high-frequency sensor observations unfolding within the reported NR period while connecting the power predictor directly to the encoder-decoder latent representation for shaft power prediction. The reconstruction and regression tasks share this latent representation under a multi-task learning objective, with reconstruction pre-training the encoder before the regression objective is introduced.

RGML_App is designed for the non-fused reports. Each NR entry provides a feature vector $x_{nr} \in \mathbb{R}^5$ and a scalar shaft power label y_{nr}. For the g-th period, the corresponding sensor sequence at hourly intervals is:

$$X^{g}_{sen} = \{x_{g,1}, \ldots, x_{g,Tg}\} \in \mathbb{R}^{(Tg \times 5)} \tag{11}$$

here, Tg denotes the number of hourly observations within the period and Tg varies across NR periods, sequences are zero-padded to a uniform length Tmax = maxg Tg with a boolean padding mask built for each sample. A mask value of one indicates padded positions to be excluded from loss computation. The

encoder takes the NR feature vector xnr ∈ $\mathbb{R}^5$ as input. The decoder reconstructs the padded sensor sequence Xgsen ∈ $\mathbb{R}$^(Tmax×5), while the regression head predicts shaft power ynr.

The model comprises an encoder, a decoder, and a regression head. The encoder is a three-layer MLP mapping NRs to latent representation z ∈ $\mathbb{R}^{128}$, with two hidden layers of 128 neurons each. We use ReLU activations, layer normalization and dropout (p = 0.2) after each layer. The decoder is a three-layer LSTM with hidden dimension 128 and inter-layer dropout (p = 0.2) reconstructing the sensor sequence by repeating z across Tmax steps and projecting the hidden states to the feature dimension:

$$H = LSTM(z \otimes 1_Tmax) \in \mathbb{R}^{(Tmax \times 128)}, \quad \hat{X}sensor = H \cdot Wout + bout \in \mathbb{R}^{(Tmax \times 5)} \quad (12)$$

The regression head is a power predictor but operates on z ∈ $\mathbb{R}^{128}$. The model is optimized using a combined loss that balances the reconstruction and regression objectives. The reconstruction loss is defined as the masked MSE over non-padded positions:

$$L_recon = [1 / \Sigma g,t(1-Mg,t)] \, \Sigma g,t \, (1-Mg,t) \cdot (1/5) \, \Sigma d_{=1}{}^{5} \, (\hat{X}sen \, g,t,d - Xsen \, g,t,d)^2 \quad (13)$$

Regression loss is MAE between predicted and actual shaft power:

$$L_reg = (1/G) \, \Sigma g_{=1}{}^{G} \, |\hat{y}g - yg| \quad (14)$$

Both losses are normalized by their values at the first training iteration to ensure balanced gradient contributions across their different scales. Training proceeds in two sequential phases. In both phases, training uses the Adam optimizer with ReduceLROnPlateau scheduler, gradient clipping (max norm 1.0) and early stopping with a minimum improvement threshold of 1×10^{-4}. The same checkpoint mechanism saves the best weights during training for a batch size 16.

Phase 1 is training for the reconstruction task only. The regression head is frozen, and only the encoder and decoder are optimized using L_recon alone, with (λ1=1.0, λ2=0.0). Adam optimizer is used with a learning rate 10^{-3}. Training is performed for up to 100 epochs with an early stopping patience of 10 epochs. The best encoder-decoder weights are restored after training. Phase 2 is joint fine-tuning, where the regression head is unfrozen, and all parameters are jointly optimized using L_RGML (λ1=λ2=0.5) with learning rate reduced to 3×10^{-4} for next 100 epochs. Figure 5 illustrates the complete RGML_App pipeline in a block diagram format.

We evaluate multiple encoder–decoder configurations using hidden and latent dimension combinations of 64/64, 128/128, 256/128, and 512/256. The 64/64 configuration reports underfitting, whereas 512/256 results in severe reconstruction overfitting without improving the regression task. Among all configurations, 128/128 consistently outperforms across all vessels, has lower fold variance, and has better convergence for phase 2. Similar to the 512/256 architecture, the 256/128 configuration also reports regression overfitting in phase 2, while a latent dimension of 512 performs worse than 256. Furthermore, we experiment with Huber loss for both the regression and reconstruction tasks, using it for regression as it has less sensitivity to outliers, but no performance improvement is observed.

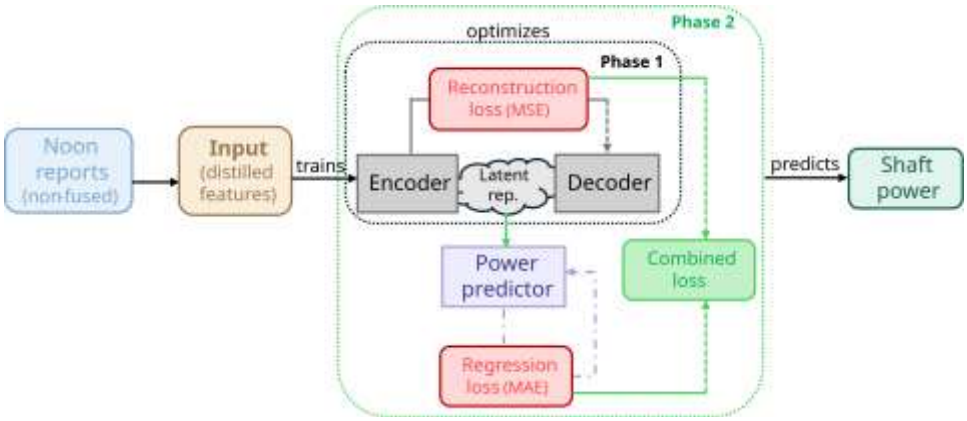


Fig. 5 Power predictions using reconstruction-guided multi-task learning: seq2seq reconstruction as an auxiliary task (Phase 1) with shared latent representation for training the power predictor (Phase 2).

6 Evaluation Design

The three proposed approaches, ALS_App, TL_App and RGML_App are evaluated across a series of experiments designed to reflect the practical challenges of working with NRs. We first formulate four research questions that frame these challenges, then describe the experimental setup developed to answer them and define the performance metrics used for the evaluation.

6.1 Research question

We address the following research questions:

- RQ1 Data sparsity challenge: Can the proposed approaches for supplementing NRs with additional data improve shaft power prediction accuracy?
- RQ2 Cross-temporal generalization: Can models trained on sparse low-frequency NRs be generalized to predict shaft power at higher temporal resolutions?
- RQ3 Minimum training data sufficiency: What amount of NR training data is required under each approach to maintain an accurate prediction of vessel shaft power?
- RQ4 Cross-fleet transferability: Can models trained on single-fleet NRs generalize to shaft power prediction in other fleets, and what is the minimum amount of training data required for the transfer?

Table 2 Detailed description of NR preprocessing schemes for assessing proposed power prediction approaches.

S	Additional data	Shaft power (p in kW) Training	Testing
S-1	Drop missing data	500< p <18k	500< p <18k
S-2	Interpolate data	500< p <18k	500< p <18k
S-3	Drop missing data	10< p	500< p
S-4	Interpolate data	10< p	500< p

6.2 Experimental setup

The proposed approaches, ALS_App, TL_App and RGML_App are examined under four schemes labeled as 'S-1', 'S-2', 'S-3' and 'S-4'. Each scheme introduces preprocessing to NRs by restricting a specific operational range of shaft power and by incorporating high-resolution weather and vessel operational data from additional sources. Defining these ranges helps investigate whether including low-power conditions within normal operating ranges can improve the model's accuracy and generalization. Data from additional sources is integrated at an hourly rate for reconstruction-guided learning and at 15-minute intervals for other approaches. Furthermore, the missing values are handled by either removing null values or by applying interpolation. The interpolation approach fills only small gaps, up to three consecutive missing points, and is applied using forward and backward filling. It increases training data volume without significantly smoothing the input data. Table 2 provides a detailed description of each scheme, with each row describing the scheme's name, handling of data from additional sources and the considered shaft power value ranges.

After selecting a scheme, vessels within each approach are trained and evaluated over their respective training and testing periods using NRs (Table 1). The use of fused or non-fused data as input is determined solely by the architectural design of the approach. Each model is trained using 10-fold cross-validation, resulting in 10 candidates for evaluation. We report the performance of the model with the lowest error on testing data, but the standard deviation is computed across the fold. The performance of each approach is compared against vessel-specific baseline models trained on non-fused NRs for low-frequency shaft power prediction.

The next experiment selects the most suitable scheme for each approach and evaluates the corresponding vessel-specific trained models for high-temporal-resolution shaft power prediction, aligned with the 15-minute sensor data frequency. Additionally, a comparative analysis is conducted against vessel-specific baseline models trained on sensor data for predicting shaft power at high frequency.

Previous experiments focus on models trained using up to one year of data. However, since not all vessels have a full year of data, we establish an upper bound on the amount of training data required under each approach to achieve and maintain robust predictive performance comparable to that obtained with one year of training data. Therefore, the selected pre-trained models are retrained from scratch to assess training data sufficiency. The retraining is performed for each approach using varying training durations, including one week, two weeks, three weeks, one month, two months, four months, six months and eight months. To ensure diversity, 10 random start-end date pairs are sampled for each training period, with ranges being independent of the vessels' testing period. The length of the testing period remains fixed at one year, consistent with earlier experiments. During training, data pre-processing is applied based on the opted scheme under each approach. The models trained on weekly data use three-fold cross-validation, those trained on one to two months of data use five-fold and models trained on data for more than two months use 10-fold cross-validation.

So far, experiments are conducted across vessels within fleet A. The next set of experiments evaluates performance across a heterogeneous fleet B. For this cross-fleet transferability, we select a vessel from fleet A that demonstrates high performance across the three proposed approaches within the chosen scheme. Models trained on this vessel are used as base models to transfer operational and weather-related insights between fleets. For TL_App, the base model is fine-tuned using the NRs of vessels in fleet B. However, for the remaining approaches, the pre-trained base model is evaluated for power predictions in fleet B with no adaptations. The evaluation period aligns with the testing period defined for vessels in fleet B. Predictive performance is compared against a baseline trained on non-fused NRs from vessels in fleet B. The prior experiment on fleet A has resulted in ten models. However, the evaluation results are reported for the best-performing model with standard deviations computed across all models. Furthermore, we also investigate the amount of training data required under each approach for accurate and reliable transfer. We evaluate the minimum training data threshold using the same experimental design employed in assessing the data sufficiency condition in fleet A. For TL_App, the base model is fine-tuned using fleet B NRs over training durations shorter than one year, while evaluation is performed over the full testing period for each vessel.

For the remaining approaches, the pre-trained fleet A models corresponding to each short training duration are directly employed in predicting power for vessels in fleet B.

6.3 Evaluation metrics

We evaluate the performance of each experiment using the following metrics:

Mean absolute percentage error (MAPE): MAPE quantifies the prediction error as a percentage of actual values and is calculated as follows:

$$MAPE = (100\%/n)\, \Sigma_{i=1}^{n} |(y_i - \hat{y}_i)/y_i| \quad (15)$$

where y is the ground truth and ŷ is the predicted value.

Mean absolute error (MAE): MAE measures the average magnitude of the prediction errors and is presented in the same units as the outcome variable. The formula used for calculation is as follows:

$$MAE = (1/n)\, \Sigma_{i=1}^{n} |y_i - \hat{y}_i| \quad (16)$$

Coefficient of determination (R^2): R^2 measures the proportion of variance in the actual data explained by the model by quantifying how well the predictions align with the ground truth. The formula used for calculation is as follows:

$$R^2 = 1 - [\Sigma_{i=1}^{n} (y_i - \hat{y}_i)^2] / [\Sigma_{i=1}^{n} (y_i - \bar{y})^2] \quad (17)$$

where ȳ is the mean of actual values (labels).

7 Results and Analysis

The experimental results for the proposed approaches are presented in this section. We first evaluate predictive accuracy within fleet A across the four data schemes, identify the best-performing scheme for each approach and compare their performance for each vessel within fleet A. Next, we examine the generalization capability of trained models (low temporal resolution) by testing their performance in predicting power at higher temporal resolutions. Subsequently, we determine the minimum training data required under each approach to sustain accurate shaft power prediction within fleet A. We conclude by assessing the ability of models trained on fleet A to generalize to power prediction in fleet B and determine the minimum amount of training data required for reliable cross-fleet transferability.

7.1 Improving predictive accuracy within a fleet: NRs vs. fused NRs

Proposed approaches are evaluated and their predictive performance are compared against a baseline model. This experiment assesses whether data fusion improves prediction accuracy for vessels in fleet A, thereby answering RQ1.

Baseline models are trained and evaluated using non-fused NRs, whereas the proposed approaches use fused NRs under four training schemes. The scheme S-1 performs best for ALS_App and RGML_App, implying that training within the operational power range without interpolation gives the highest accuracy. For the baseline and TL_App, S-2 performs best, implying that interpolation helps but extending the power range provides no further benefit.

Models under the best-performing scheme for each approach, including the baseline are compared to assess their relative strengths and limitations. Table 3 presents the evaluation metrics from the most efficient scheme identified for each approach for fleet A. The first column indicates the approach, followed by the vessel name and the corresponding evaluation metrics. For each vessel, the best metric combination across approaches is highlighted in bold.

Table 3 Power prediction at low frequency: comparative analysis of approaches for fleet A.

Approach	V	R^2	MAPE	MAE
Baseline	A-1	0.69±0.5	24.57±7.7	838.0±520.4
	A-2	0.21±0.3	30.40±7.2	1342.2±259.7
	A-3	0.45±0.1	19.86±2.5	529.5±49.5
	A-4	0.35±0.3	17.07±4.8	831.0±215.5
	A-5	-12.28±10.7	20.28±3.0	923.0±96.1
	A-6	-0.20±0.8	34.09±9.8	1702.3±370.0
	A-7	0.66±0.2	19.26±5.6	552.6±113.6
	A-8	0.12±0.2	26.09±1.9	897.7±104.7
ALS_App	A-1	0.66±0.1	17.95±3.4	680.3±154.5
	A-2	-1.19±0.3	33.43±5.5	1857.6±265.8
	A-3	0.69±0.0	18.76±1.7	515.0±23.0
	A-4	0.83±0.4	10.10±6.8	515.2±298.6
	A-5	0.42±0.0	13.47±1.3	646.1±27.3
	A-6	0.33±0.2	20.17±2.5	1089.2±121.4
	A-7	-1.39±2.8	10.82±4.0	558.1±214.7
	A-8	-0.07±0.3	16.76±2.4	754.3±192.1
TL_App	A-1	0.83±0.0	16.44±0.8	616.2±19.2
	A-2	0.65±0.0	18.71±3.0	798.2±66.8
	A-3	0.37±0.2	22.51±4.7	576.6±105.7
	A-5	-14.36±1.0	17.30±0.4	768.3±33.0
	A-6	0.46±0.0	21.12±0.0	1078.1±0.2
	A-7	0.59±0.8	22.95±14.3	647.2±463.9
	A-8	0.31±0.0	19.02±0.5	731.3±7.3
RGML_App	A-1	0.66±0.2	23.69±5.0	933.6±211.5
	A-2	-0.20±0.3	32.15±3.3	1600.7±157.5
	A-3	0.34±0.0	21.76±1.2	604.1±26.0
	A-4	0.69±0.1	14.04±1.3	744.2±68.0
	A-5	0.39±0.1	13.32±1.3	650.6±60.8
	A-6	0.26±0.1	27.05±2.1	1303.4±108.8
	A-7	0.57±0.1	20.25±1.8	613.5±52.8
	A-8	0.08±0.1	26.82±1.7	987.3±60.0

Overall, the baseline model demonstrates the data sparsity issue as mentioned in RQ1. The results have high errors, low R^2 scores and high standard deviations, indicating inconsistent learning. The model establishes the best performance that can only be achieved using non-fused NRs. The best result is reported for A-7, worst for A-5 and A-6. The performance results represent the fundamental limitation of sparse, low-frequency NRs data. The baseline should be considered only when computational resources are constrained. ALS_App provides significant improvement in predictive performance for vessels with diverse operational data with lower standard deviations as compared to baseline. A-4 achieves the highest R^2 and A-5 improves from a negative to a positive R^2 value. ALS_App outperforms the baseline for several vessels and is best suited for vessels with consistent NR histories, particularly when minimizing MAPE is the primary objective. It should be avoided for vessels with irregular data.

TL_App shows the most consistent performance across vessels and is the only approach that effectively models A-2. A-1 achieves its best performance under this approach. However, the source model fails to transfer relevant representations for A-5 and under performs for A-3, indicating that TL_App may introduce bias for vessels having sufficient own vessel-specific data. The approach is best suited for data-sparse vessels and could be the default choice for new vessels with limited historical data. RGML_App shows a relatively uniform performance across vessels. Standard deviations are lower than in the baseline and ALS_App, indicating that the reconstruction objective regularizes the latent representation and reduces overfitting. This approach is well suited for generalization across a fleet when there is no reliable source vessel for TL. It provides a practical middle ground and is particularly suited for scenarios where data arrives sequentially and the reconstruction objective can exploit temporal patterns in NRs.

Answering RQ1, all three approaches demonstrate measurable improvements for most vessels, confirming that incorporating additional data can overcome the limitations of sparse NRs. However, the results also reveal that no single approach universally overcomes data sparsity across all vessels, indicating that an approach selection should be tailored to vessel-specific data.

7.2 Generalizing to higher temporal resolution: fused NRs vs. sensor data

In this section, trained models under the proposed approaches are evaluated for predicting shaft power at high frequency. This evaluation examines the predictive performance of models pre-trained on low-frequency NRs for high-temporal-resolution prediction tasks. The evaluation results are compared against baseline models trained on sensor data over the same time period. The baseline model has the same architecture as the power predictor. The baseline model is trained and evaluated under the same four schemes and the best performance is observed for S-1. In this experiment, we assess the robustness and generalization capability of the proposed approaches, thereby addressing RQ2.

Table 4 presents the evaluation metrics for the approaches, including the baseline model across fleet A. The first column indicates the selected model, followed by the vessel name and the corresponding metric. For each vessel, the best metric combination across the schemes is highlighted in bold.

Table 4 Power prediction at high frequency: comparative analysis of approaches for fleet A.

Approach	V	R^2	MAPE	MAE
Baseline	A-1	0.60±0.0	19.16±1.7	939.1±53.7
	A-2	0.75±0.0	15.11±0.9	712.2±39.3
	A-3	0.48±0.1	22.75±1.4	563.7±32.9
	A-4	0.82±0.0	8.22±1.2	473.5±64.8
	A-5	0.38±0.1	13.27±0.6	619.4±31.7

Approach	V	R^2	MAPE	MAE
	A-6	0.01±0.1	27.53±1.5	1411.9±84.5
	A-7	0.68±0.1	18.22±1.7	557.3±41.8
	A-8	-1.34±0.1	33.16±1.0	1683.5±61.5
ALS_App	A-1	0.75±0.2	19.78±6.8	735.3±159.8
	A-2	-0.57±0.3	32.92±11.3	1644.7±332.4
	A-3	0.43±0.0	23.62±1.3	599.2±22.8
	A-4	0.80±0.4	8.76±5.5	528.8±213.0
	A-5	0.48±0.0	13.30±0.4	602.0±13.9
	A-6	0.25±0.2	22.93±2.5	1208.7±117.8
	A-7	0.63±0.1	18.86±2.4	580.1±55.4
	A-8	-0.02±0.2	20.86±1.4	985.1±107.7
TL_App	A-1	0.84±0.0	16.78±0.8	605.5±18.6
	A-2	0.74±0.0	17.15±3.5	733.0±74.2
	A-3	0.33±0.2	29.88±5.6	669.4±120.1
	A-5	0.03±0.0	18.76±0.3	786.8±7.9
	A-6	0.09±0.0	27.60±0.0	1299.3±0.4
	A-7	0.56±0.9	24.21±15.3	674.6±509.2
	A-8	0.41±0.0	18.95±0.8	794.1±14.9
RGML_App	A-1	0.47±0.1	27.15±2.6	1133.0±112.1
	A-2	0.65±0.0	21.15±1.1	890.2±34.2
	A-3	0.36±0.1	23.36±3.0	617.6±71.7
	A-4	0.66±0.1	12.84±1.3	740.9±87.6
	A-5	0.21±0.1	17.40±0.7	804.4±30.2
	A-6	0.01±0.2	26.17±1.5	1382.0±96.1
	A-7	0.46±0.1	23.18±1.6	684.1±38.6
	A-8	-0.80±0.3	29.20±2.7	1432.1±149.7

The results show that ALS_App reports the most consistent generalization and outperforms the sensor-based baseline on at least two of the three metrics for vessels A-1, A-5, A-6, and A-8. In fact, for A-6 and A-5, the best performance is achieved within this approach. TL_App reports the best result in the experiment for A-1, surpassing the baseline on all three metrics ($R^2 = 0.84$, MAPE = 16.78%, MAE = 605.5 kW). For A-8, a negative baseline $R^2 = -1.34$ improves to 0.41, representing the biggest change in R^2 observed across all approaches. This demonstrates that pre-training on NRs, followed by fine-tuning for sister vessels offers better representation than training directly on sensor data. TL_App is the only approach having a positive R^2 across all vessels, suggesting its ability to bridge the temporal gap between NRs and sensor observations. RGML_App proves to be the least competitive approach under this experiment. It fails

to match the baseline on any metric, with the exception of A-6 and A-8. It suggests that the reconstruction objective was ineffective in the high-frequency prediction task. Post-hoc analysis of the latent space (Appendix A, Table A2) shows that operational features (draft, wave height) are reconstructed with high fidelity, while directional features are captured less accurately. This disparity provides a possible explanation for the approach's limited improvements in high-frequency prediction.

Answering RQ2, models trained on NRs can generalize to higher temporal resolutions, though the extent of generalization varies depending on the vessel and approach. TL_App is the most capable for cross-temporal generalization, ALS_App shows the broadest applicability and outperforms the baseline on at least two metrics for four vessels. However, no approach has achieved complete generalization across all vessels and RGML_App shows limited success in cross-temporal transfer. Additionally, we observe that for vessels with poor baseline performance (A-6, A-8), all three approaches consistently attempt to reduce the performance gap relative to the baseline. This suggests that NRs augmented with data from additional sources can effectively regularize power prediction accuracy when sensor data training is insufficient.

7.3 Bridging the temporal gap: synthesizing NR and sensor-level predictive accuracy

Sections 7.1 and 7.2 evaluate the proposed approaches separately at low frequency (NRs) and high frequency (sensor data). This section synthesizes both RQ1 and RQ2 to assess whether the improvements observed at NR resolution are retained when the same approaches are evaluated against sensor-level ground truth. Only R^2 is used as the comparison criterion because it is directly comparable across both resolutions. MAE and MAPE are scaled to the shaft power distribution at each resolution and are therefore not directly comparable. Figure 6 summarizes vessel-level outcomes for each approach at both resolutions. Panel (a) reports performance at low frequency against the non-fused NR baseline and classifies each vessel as 'improves/full bridges' (R^2fused $\geq$ R^2NRbaseline) or 'no improvement' (otherwise). Panel (b) reports performance at high frequency against two references, the non-fused NR baseline and the sensor baseline. Thus, we have three categories as 'improves/full bridges', where our approaches matches or exceeds the sensor baseline (R^2fused > R^2sensorbaseline), 'partial improvement', where they exceed the non-fused NR baseline but remains below the sensor baseline (R^2NRbaseline < R^2fused < R^2sensorbaseline) and 'no improvement', where it fails to surpass the NR baseline (R^2fused $\leq$ R^2NRbaseline).

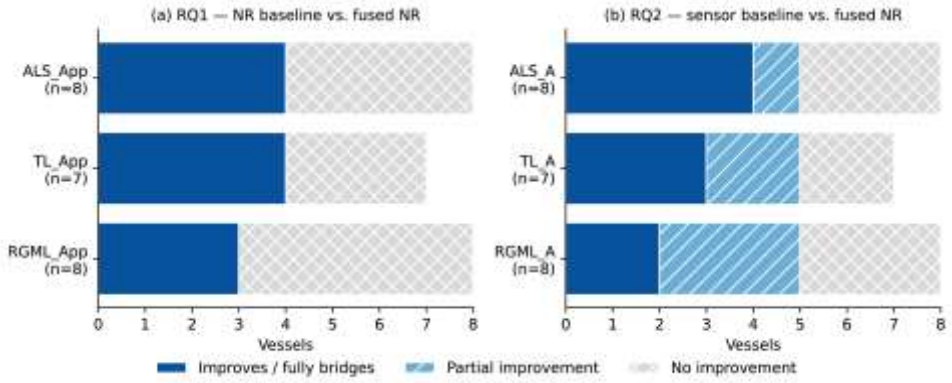


Fig. 6 Vessel-level outcomes per approach for (a) Improvement over NR baseline at low frequency (RQ1) and (b) Outcomes relative to sensor baselines at high frequency (RQ2): fully bridging the gap, partial improvement over the baseline, or no improvement.

The results indicate that ALS_App generalizes most consistently across resolutions, TL_App achieves similar improvements at low frequencies and is less effective at reducing the sensor-level discrepancy. RGML_App demonstrates stability within a single resolution, yet performs worst at bridging the gap between resolutions.

7.4 Maintaining predictive accuracy: one year vs. shorter training periods

The vessel-specific models under the proposed approaches are originally trained over one year of data. These models are retrained from scratch for varying shorter training duration. This evaluation examines the performance of each approach as the size of NRs increases, with the goal of achieving reliable shaft power predictions, thereby answering RQ3.

However, to evaluate training data sufficiency, we consider only the top four best-performing vessels from each approach. Figures 7, 8 and 9 illustrate the trend in error rate and model fit for ALS_App, TL_App and RGML_App, respectively, with increasing amount of training data. Each subplot shows the performance of individual vessels, with the training period on the x-axis, change in MAPE on the left y-axis and change in R^2 on the right y-axis.

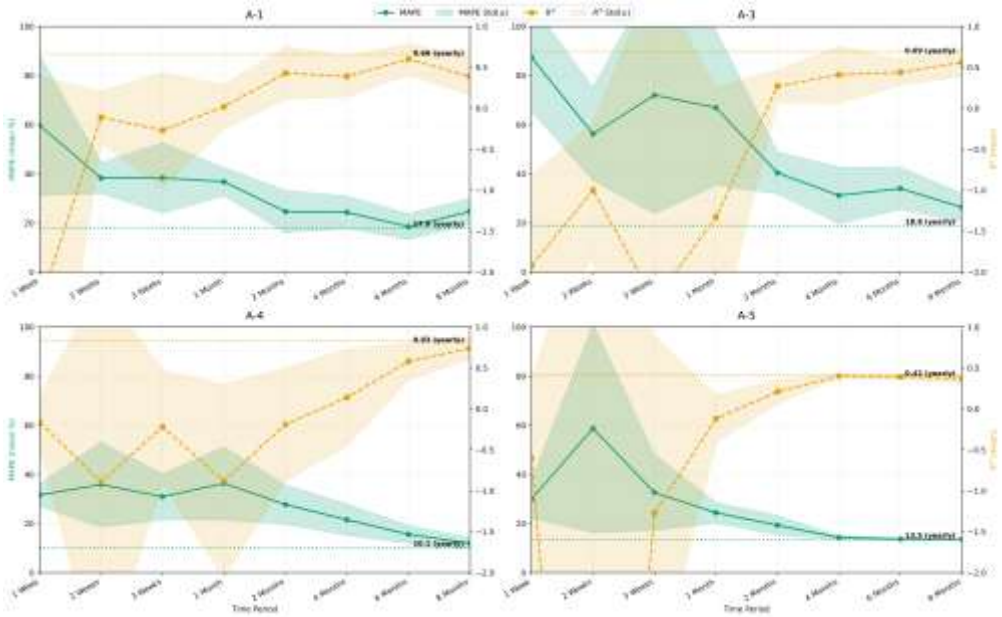


Fig. 7 Aggregate loss supervision: impact of training data volume on predictive accuracy relative to yearly baseline.

Trend for ALS_App: (Figure 7) We observe a consistent pattern across four vessels, an unstable performance with shorter periods, which is followed by strong convergence with longer durations. Based on the convergence behavior across all vessels, a minimum of six months of training data is required. A-1 requires at least two months for a consistently positive $R^2 = 0.43$. Its MAPE approaches the yearly benchmark by six months. A-3 shows high instability in early weeks and shows gradual improvement from two months onward and reporting $R^2 = 0.56$ at eight months. A-5 delivers the most stable performance, with negligible standard deviation and shows a positive $R^2 = 0.21$ at two months. A-4 provides the best fit and reaches stable convergence from approximately two months, but it is unstable until six months.

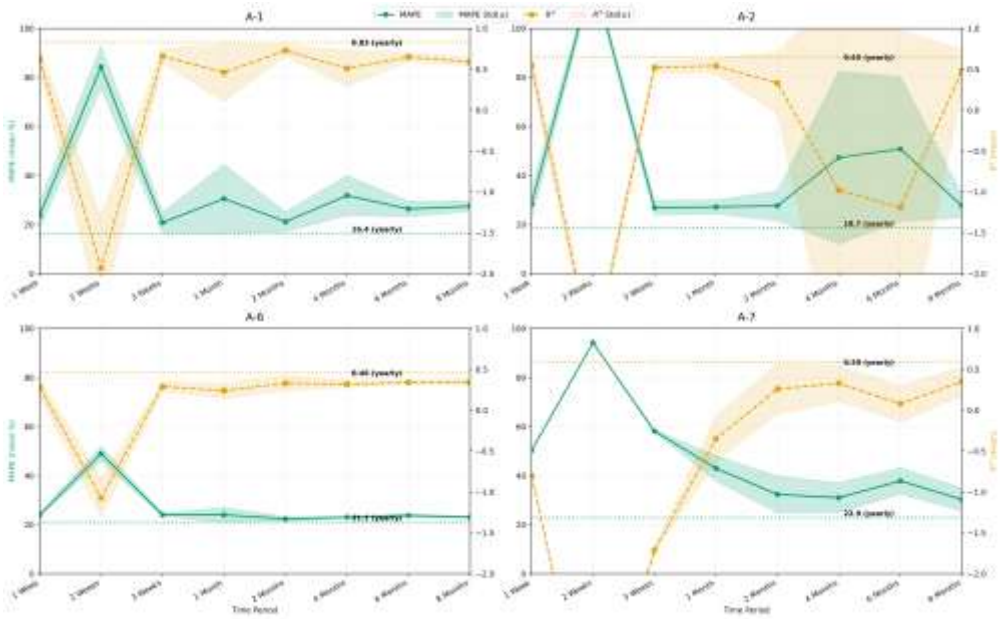


Fig. 8 Transfer Learning: impact of training data volume on predictive accuracy relative to yearly baseline.

Trend for TL_App: (Figure 8) This approach is the most efficient, as three vessels achieve a positive R^2 from one week. For reliable predictions across all vessels, the recommended optimal training period is two months. A-1 reports the best results, achieving a positive R^2 within the first week and reaching a peak value of 0.73 after two months. A-2 reports a positive R^2 from week one, but performance deteriorates after two months. A-6 is most consistent with low variance, reaching $R^2 = 0.29$ at one week. A-7 requires more training data, with R^2 improving to 0.347 after eight months.

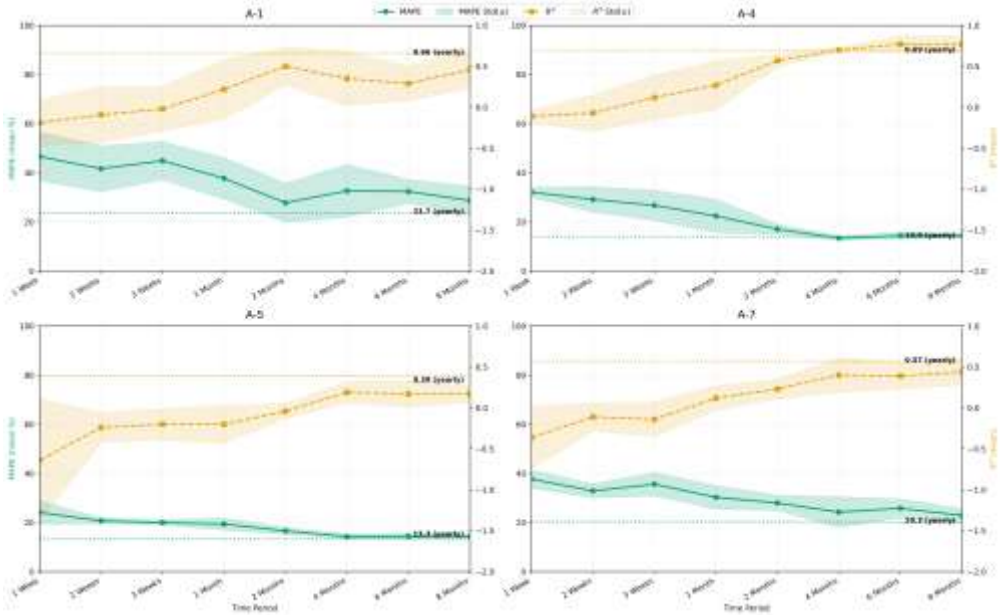


Fig. 9 Reconstruction-guided multi-task learning: impact of training data volume on predictive accuracy relative to yearly baseline.

Trend for RGML_App: (Figure 9) All vessels show consistent improvement with increasing training data. The minimum training period is four months. A-1 reaches a positive R^2 after one month and stabilizes at

0.46 by eight months. A-4 performs best, achieving a positive R^2 in three weeks with a low standard deviation. A-7 shows a positive model fit from one month and improves steadily til eight months. A-5 is the most data-demanding vessel, with R^2 turning positive only after four months.

Answering RQ3, if training data is limited TL_App is the most data-efficient, followed by RGML_App and ALS_App. But, as the training dataset size increases, ALS_App has the best model fit, followed by RGML_App and then TL_App. TL_App shows the highest variance, RGML_App has the most consistent improvement and ALS_App only convergences to reliable predictions after six months.

7.5 Generalizing and maintaining predictive accuracy across fleets: NRs vs. fused NRs

The proposed approaches have been previously evaluated exclusively on vessels in fleet A. We now investigate whether the knowledge learned from fleet A can generalize to a different fleet B, thereby addressing RQ4. The cross-fleet transferability assessment is conducted using vessel A-4 from fleet A. A-4 has outperformed across all experiments within each approach.

The predictive performance of the proposed approaches is compared against a baseline model trained using non-fused NRs from fleet B. Table 5 presents the cross-fleet transferability evaluation metrics. For each vessel, the best metric combination is highlighted in bold. Results from only the best-performing scheme are reported in the table. S-1 performs better than the other schemes in most approaches, but S-2 gives better results for ALS_App.

Similar to the data sparsity issue in RQ1, the baseline model trained solely on non-fused NRs is characterized by high MAPE and MAE with low or negative R^2. Overall, the results suggest that relying on non-fused NRs provides unreliable predictions. ALS_App achieves the most consistent cross-fleet performance, improving all three metrics for four vessels relative to the baseline. B-4 performs best (R^2 = 0.80, MAPE = 15.25%, MAE = 753.0 kW) with the greatest improvement over the baseline. B-5 improves moderately (R^2 = 0.57) and has half of the standard deviation scores as compared to the baseline and is highly stable. However, B-7 remains an outlier, indicating that the approach has converged to a suboptimal solution.

TL_App has the highest prediction stability with near-zero cross-fold variance. It achieves the best performance for B-1, B-3 and B-7. The biggest improvement is observed for B-7 (R^2 = 0.30, MAPE = 16.18%), where TL_App recovers B-7 performance from baseline failure. B-5 is the only vessel that regresses on all metrics, suggesting incompatibility with A-4. RGML_App shows the weakest cross-fleet transfer with limited improvement over the baseline. R^2 for B-1 falls from 0.70 to -0.15 and B-3 reports the highest MAE (3062.4 kW) for a vessel in this experiment. B-7 reports a positive R^2 value, but it is the lowest across approaches. RGML_App is not applicable for cross-fleet transfer.

We further determine the minimum amount of training data required by TL_App (S-1 scheme) and ALS_App (S-2 scheme) to achieve performance comparable to one year of data. Following the RQ3 experimental design, we evaluate the top four performing vessel under TL_App and ALS_App. For ALS_App, the pre-trained A-4 models from RQ3 is used at each duration. For TL_App, we fine-tune the corresponding A-4 base model using NRs from fleet B before predicting shaft power. Figure 10 and 11 illustrates how predictive performance changes for both TL_App and ALS_App as training data increases across the selected vessels.

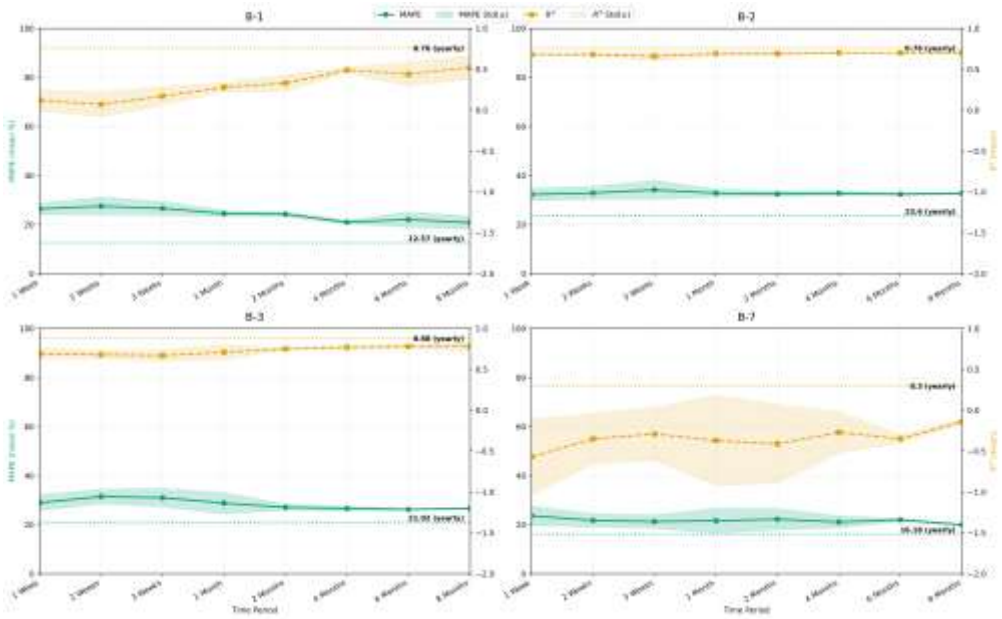


Fig. 10 Transfer learning across fleets: impact of training data volume on predictive accuracy relative to yearly baseline.

TL_App requires nearly one year of source-vessel fused NRs to reach the benchmark performance (Table 5), with two months providing a stable but suboptimal lower bound. B-3 is the most data-efficient, as show early and stable convergence. B-1 shows slower and unstable convergence, while B-2 has early plateauing. B-7 is the most data-demanding and has negative R^2 up to eight months. In the case of ALS_App, all vessels show poor generalization with considerable instability persisting through three weeks. From one month onward, MAPE improves gradually across all vessels. Positive R^2 is reported at six months for B-4 (0.14), B-5 (0.37) and B-6 (0.20), while B-2 reports a positive R^2 only at eight months (0.34). Although variance recedes by eight months, all vessels remain below their yearly benchmarks. One year of training data is required for reliable cross-fleet predictions, with six months serving as a lower bound for achieving partial performance. In comparison to TL_App, ALS_App requires more data for cross-fleet transferability.

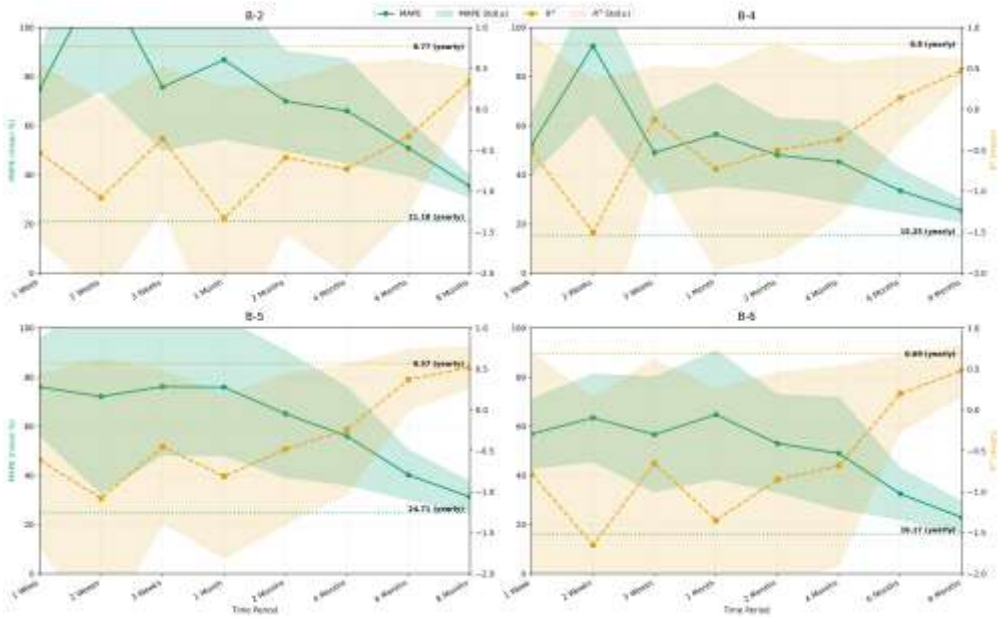


Fig. 11 Aggregate loss supervision across fleets: impact of training data volume on predictive accuracy relative to yearly baseline.

Answering RQ4, cross-fleet transferability is achievable using fused NRs from a single fleet. In general, a full year of fused NRs from the source vessel is required. TL_App is more robust for general cross-fleet applications and will be the preferred approach when data is limited.

Table 5 Power prediction at low frequency across fleets: comparative analysis of approaches for fleet B.

Approach	V	R^2	MAPE	MAE
Baseline	B-1	0.70±0.15	13.96±7.6	1119.0±250.3
	B-2	0.68±0.04	32.77±3.8	1049.4±83.9
	B-3	0.86±0.97	19.53±18.9	1256.4±1842.4
	B-4	0.57±0.03	24.30±1.4	1186.0±62.9
	B-5	0.49±0.11	29.99±3.1	1472.5±131.5
	B-6	0.60±0.14	25.75±3.0	1093.3±131.9
	B-7	-0.44±0.23	22.80±1.4	1712.5±126.9
ALS_App	B-1	0.50±0.45	17.58±10.1	1587.3±729.4
	B-2	0.77±0.14	21.18±5.6	782.2±219.7
	B-3	0.62±0.11	21.53±4.7	1885.7±261.7
	B-4	0.80±0.10	15.25±4.1	753.0±207.2
	B-5	0.57±0.05	24.71±3.1	1077.1±84.9
	B-6	0.69±0.20	16.17±4.7	775.9±226.3
	B-7	0.18±0.07	29.59±1.5	1840.8±69.4
TL_App	B-1	0.76±0.01	12.57±0.3	998.2±7.7
	B-2	0.76±0.00	23.60±0.8	880.0±22.2
	B-3	0.88±0.02	21.02±4.6	1140.8±129.8
	B-4	0.63±0.00	21.72±3.1	1080.7±45.9
	B-5	0.49±0.00	33.06±0.9	1532.4±7.1
	B-6	0.68±0.00	21.93±0.2	968.2±3.3
	B-7	0.30±0.05	16.18±0.4	1132.6±45.0
RGML_App	B-1	-0.15±0.11	36.04±2.0	2732.5±151.0
	B-2	0.67±0.03	41.09±5.2	1142.9±70.4
	B-3	0.06±0.07	36.92±1.7	3062.4±106.1
	B-4	0.49±0.06	24.17±1.6	1357.2±79.1
	B-5	0.53±0.03	24.31±1.9	1282.5±39.8
	B-6	0.44±0.06	22.63±1.1	1139.5±79.1

Approach	V	R^2	MAPE	MAE
	B-7	0.19±0.07	27.85±1.3	1823.0±88.1

7.6 Approach selection: practical guidelines

The experimental results demonstrate that no single modeling approach is universally optimal. Instead, approach selection depends primarily on the amount of historical NR data available and whether there exist a period for which vessel-specific NRs can be collected before deploying a power prediction model. Figure 12 presents flowcharts summarizing the recommended decision process facilitating practical applications.

For a vessel with no historical data, the recommended approach depends on whether any vessel-specific NR data can be collected prior to power prediction modeling, rather than on the similarity between other vessels. If no NR exists, ALS_App with a pre-trained model (across the fleet) without any vessel-specific training can be deployed (Section 7.5). When even a limited amount of vessel-specific NR data can be collected, TL_App is the recommended approach, requiring approximately one month of data for vessels within the same fleet (Section 7.4) and up to one year for cross-fleet deployment (Section 7.5). For vessels with two to four months of historical NR, RGML_App provides the best balance of performance and stability. After six to eight months, ALS_App is recommended because it consistently achieves the highest predictive accuracy (Section 7.4).

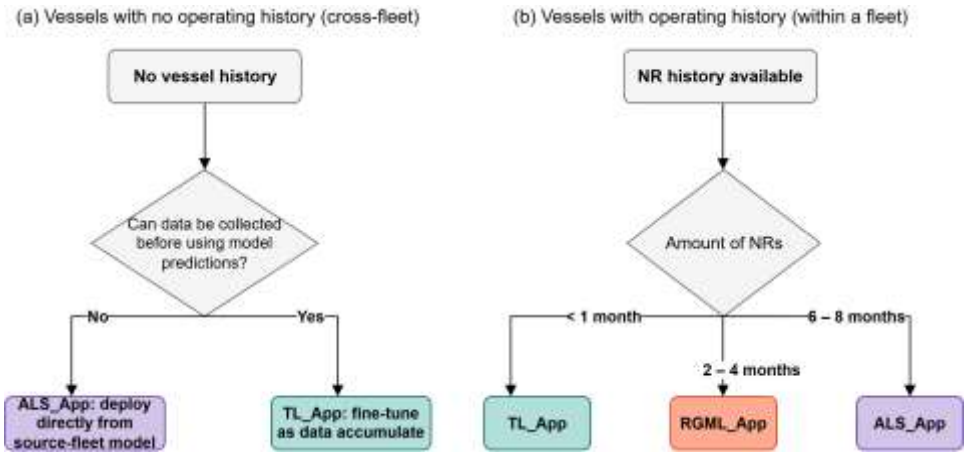


Fig. 12 Recommended approach selection: practical workflow based on (a) data collection feasibility for vessels with no operating history, and (b) amount of historical NRs available for vessels with existing historical NRs.

For cross-fleet applications, TL_App is the preferred approach provided an appropriate source-target vessel can be identified. We investigate whether a quantitative compatibility index (Section 5.3) for a source-target vessel relationship can further refine the above guidance process. However, structural and operational similarities report opposite relationships with TL_App performance. Structural similarity improves TL_App performance, whereas operational similarity reduces it (structural: $\rho = -0.81$, $p = 0.03$ and operational: $\rho = +0.78$, $p = 0.04$). As a result, combining the two similarities with equal weighting weakened the relationship, making it non-significant ($\rho = -0.63$, $p = 0.13$). This indicates that our data do not support a reliable compatibility index. On the contrary, a different measure better explains the results. It is the target vessel's baseline (non-fused NRs) predictability from its own NR history before transfer learning. As evident from Table 5, fleet B vessels with higher baseline R^2 scores also achieve higher TL_App R^2 scores ($\rho = 0.99$, $p < 0.001$). We therefore recommend using the target vessel's baseline predictability rather than the compatibility score as the practical criterion for approach selection.

8 Conclusion

This study investigates shaft power prediction using NRs, one of the primary data sources for most vessels despite their low temporal resolution. We propose three deep learning approaches: ALS_App, TL_App, and RGML_App to fuse NRs with publicly available data under different data availability conditions. We evaluate them across two heterogeneous fleets, four preprocessing schemes and two temporal resolutions. All three approaches demonstrate substantial improvements over the baseline for most vessels, validating that fusing publicly available data with NRs can mitigate the effects of data sparsity.

Two important findings emerge as this study's core contribution. First, the limitations of sparse noon reports can be addressed by restructuring supervision (ALS_App), transferring representation (TL_App), or introducing an auxiliary reconstruction objective (RGML_App), each improving per-instance learning from NR data. However, the most effective approach depends on the vessel and data characteristics. Second, the amount of historical NR data is a more reliable criterion for selecting an approach than vessel similarity. We had expected a structural and operational compatibility index to guide the approach selection, but it showed no significant relationship with model performance. Instead, the target vessel's own baseline performance (with non-fused NRs) consistently predicted which approach would be most effective, specifically in case of TL_App. Practically, this means operators do not need to search for a "sister vessel" before deploying a source model, rather a simple baseline assessment of the target vessel's own sparse NRs is a better guide when selecting a prediction approach.

Overall, the results support selecting a power prediction approach according to data availability rather than adopting a single model for all vessels. TL_App performs best for data-sparse vessels and RGML_App is a robust alternative when two to four months of NRs are available. ALS_App is most suitable when at least six months of NRs are available. These findings suggest that the fusion strategy should be matched to the available data rather than the model architecture. For cross-fleet deployment, TL_App is the preferred approach, reporting the most accurate and consistent performance after adaptation to the target fleet. ALS_App also supports cross-fleet prediction and, unlike TL_App, requires no training data from the target fleet. Predictions are made directly using the pre-trained source model without target fleet adaptation, making ALS_App suitable for new fleets with no historical data. In contrast, RGML_App is not suitable for cross-fleet transfer.

Limitations and future work: This work also has certain limitations that should be acknowledged. The study is limited to bulk carrier and tanker vessels and generalization to other vessel categories is not evaluated. For TL_App, only last-layer fine-tuning (TL-based feature extraction strategy) has been evaluated. Alternative strategies, such as partial or full fine-tuning [27] may result in different performance trade-offs and require further investigation. Data fusion between NR and public data relies on linear (AIS) and trilinear (CMEMS/ERA5) interpolation, which assumes smooth vessel kinematics between observations. It is a weaker assumption during maneuvering, given known AIS quality issues [6, 7]. The robustness check (Appendix A, Table A1) confirms that upsampling preserves the aggregate NR signal but does not establish instance-level realism. Future work could assess performance by using more physically constrained trajectory reconstruction mechanism such as Kalman filtering. Cross-fleet transferability is also evaluated using a single source vessel (A-4). Since transfer performance is better explained by target-vessel baseline predictability than by our compatibility index, future work should evaluate source selection across multiple candidate vessels. Additionally, this study does not investigate whether feature importance varies across fleet, (bulk carriers and oil tankers). Future work could use SHAP [11] to evaluate fleet-specific feature importance and determine whether category-specific feature sets improve predictive accuracy.

Under cross-fleet settings, exploring adaptive fine-tuning strategies to reduce the data requirements of ALS_App represents a promising research direction. In the cross-fleet data sufficiency experiment, TL_App fine-tunes the source model corresponding to each source data duration, ensuring a fair comparison with ALS_App. Future work should evaluate whether fine-tuning a fully pre-trained one-year source model

improves cross-fleet performance. Also, investigating whether physics-informed constraints can further stabilize predictions under limited training data remains an important direction for further study. Combining the complementary strengths of the proposed approaches represents a promising research direction. Hybrid architectures integrating TL_App's pre-trained backbone with the aggregate loss objective or the reconstruction-guided encoder could improve the robustness and accuracy of shaft power prediction in both within and cross-fleet settings, particularly for vessels where individual approaches currently show mixed results.

Data and code availability: The data and code used in this paper are confidential, as they belong to the industrial partner and therefore cannot be made publicly available.

Acknowledgment: This study has been funded by the Research Council of Norway under grant agreement No. 346603, the GASS project. We used data from the E.U. Copernicus Marine Service Information. The authors thank Nassim Belmecheri for valuable discussions that contributed to the development of the methodology.

Author contribution statement: Akriti Sharma: conceptualization, methodology, software, validation, visualization, writing - original draft, writing - review & editing. Dogan Altan: writing - original draft, writing - review & editing. Joachim Haga: conceptualization, data curation, writing - review & editing. Glenn Lines: methodology, writing - review & editing. Dusica Marijan: funding acquisition, writing - review & editing. Arnbjørn Maressa: resources.

References

[1] Ferrari, E., Christidis, P., Bolsi, P.: The impact of rising maritime transport costs on international trade: Estimation using a multi-region general equilibrium model. Transportation Research Interdisciplinary Perspectives 22, 100985 (2023)

[2] Marijan, D., Mohammed, H.H., Zaman, B.: Estimation and optimization of ship fuel consumption in maritime: review, challenges, and future directions. Journal of Marine Science and Technology, 1–23 (2025)

[3] Yang, D., Wu, L., Wang, S., Jia, H., Li, K.X.: How big data enriches maritime research—a critical review of Automatic Identification System (AIS) data applications. Transport Reviews 39(6), 755–773 (2019)

[4] Rio, M.-H., Mulet, S., Picot, N.: Beyond GOCE for the ocean circulation estimate: Synergetic use of altimetry, gravimetry, and in situ data provides new insight into geostrophic and Ekman currents. Geophysical Research Letters 41(24), 8918–8925 (2014)

[5] Perera, L.P., Soares, C.G.: Weather routing and safe ship handling in the future of shipping. Ocean Engineering 130, 684–695 (2017)

[6] Altan, D., Marijan, D., Kholodna, T.: SafeWay: Improving the safety of autonomous waypoint detection in maritime using transformer and interpolation. Maritime Transport Research 4, 100086 (2023)

[7] Liang, M., Su, J., Liu, R.W., Lam, J.S.L.: AISClean: AIS data-driven vessel trajectory reconstruction under uncertain conditions. Ocean Engineering 306, 117987 (2024)

[8] Nguyen, D., Fablet, R.: A transformer network with sparse augmented data representation and cross entropy loss for AIS-based vessel trajectory prediction. IEEE Access 12, 21596–21609 (2024)

[9] Nguyen, S., Gadel, M., Wang, K., Li, J., Zhang, X., Kong, S.-C., Fu, X., Qin, Z.: Maritime decarbonization through machine learning: A critical systematic review of fuel and power prediction models. Cleaner Logistics and Supply Chain 14, 100210 (2025)

[10] Kiouvrekis, Y., Gkirtzou, K., Zikas, S., Kalatzis, D., Panagiotakopoulos, T., Lajic, Z., Papathanasiou, D., Filippopoulos, I.: An Explainable Machine Learning Approach for IoT-Supported Shaft Power Estimation and Performance Analysis for Marine Vessels. Future Internet 17(6), 264 (2025)

[11] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems 30 (2017)

[12] Kim, D., Handayani, M.P., Lee, S., Lee, J.: Feature Attribution Analysis to Quantify the Impact of Oceanographic and Maneuverability Factors on Vessel Shaft Power Using Explainable Tree-Based Model. Sensors 23(3), 1072 (2023)

[13] Laurie, A., Anderlini, E., Dietz, J., Thomas, G.: Machine learning for shaft power prediction and analysis of fouling related performance deterioration. Ocean Engineering 234, 108886 (2021)

[14] Vorkapić, A., Radonja, R., Babić, K., Martinčić-Ipšić, S.: Machine learning methods in monitoring operating behaviour of marine two-stroke diesel engine. Transport 35(5), 474–485 (2020)

[15] dos Santos Ferreira, R., de Lima, J.V.P., Caprace, J.-D.: Comparative analysis of machine learning prediction models of container ships propulsion power. Ocean Engineering 255, 111439 (2022)

[16] Kim, D., Lee, S., Lee, J.: Data-Driven Prediction of Vessel Propulsion Power Using Support Vector Regression with Onboard Measurement and Ocean Data. Sensors 20(6), 1588 (2020)

[17] Uyanık, T., Yalman, Y., Kalenderli, Ö., Arslanoğlu, Y., Terriche, Y., Su, C.-L., Guerrero, J.M.: Data-driven approach for estimating power and fuel consumption of ship: A case of container vessel. Mathematics 10(22), 4167 (2022)

[18] Liang, Q., Tvete, H.A., Brinks, H.W.: Prediction of vessel propulsion power using machine learning on AIS data, ship performance measurements and weather data. In: Journal of Physics: Conference Series, vol. 1357, p. 012038 (2019). IOP Publishing

[19] Parkes, A.I., Savasta, T.D., Sobey, A.J., Hudson, D.A.: Efficient vessel power prediction in operational conditions using machine learning. In: Practical Design of Ships and Other Floating Structures, pp. 350–367. Springer (2019)

[20] Parkes, A.I., Savasta, T.D., Sobey, A.J., Hudson, D.A.: Power prediction for a vessel without recorded data using data fusion from a fleet of vessels. Expert Systems with Applications 187, 115971 (2022)

[21] Karagiannidis, P., Themelis, N.: Data-driven modelling of ship propulsion and the effect of data pre-processing on the prediction of ship fuel consumption and speed loss. Ocean Engineering 222, 108616 (2021)

[22] Mohammed, H.H., Marijan, D., Maressa, A.: Physics-Informed Machine Learning for Vessel Shaft Power and Fuel Consumption Prediction: Interpretable KAN-Based Approach. In: 2025 IEEE International Conference on Big Data (BigData), pp. 4653–4662 (2025)

[23] Altan, D., Mohammed, H.H., Lines, G.T., Marijan, D., Maressa, A.: Physics-Guided Neural Network-Based Shaft Power Prediction for Vessels. In: 2025 IEEE International Conference on Big Data (BigData), pp. 2715–2723 (2025)

[24] Parkes, A.I., Sobey, A.J., Hudson, D.A.: Physics-based shaft power prediction for large merchant ships using neural networks. Ocean Engineering 166, 92–104 (2018)

[25] Bourchas, O., Papalambrou, G.: Physics informed neural networks for vessel main engine power prediction. Ocean Engineering 332, 121344 (2025)

[26] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljacic, M., Hou, T.Y., Tegmark, M.: KAN: Kolmogorov–Arnold Networks. In: The Thirteenth International Conference on Learning Representations (2025)

[27] Luo, X., Zhang, M., Han, Y., Yan, R., Wang, S.: Ship fuel consumption prediction based on transfer learning: Models and applications. Engineering Applications of Artificial Intelligence 141, 109769 (2025)

[28] Mavroudis, S., Tinga, T.: Application of transfer learning on physics-based models to enhance vessel shaft power predictions. Ocean Engineering 323, 120540 (2025)

[29] Fan, A., Sun, S., Hu, Z., Vladimir, N., Mao, W.: Enhancing transfer learning strategies for ship fuel consumption prediction under data scarcity. Ocean Engineering 351, 124398 (2026)

[30] Guo, Y., Wang, Y., Zhou, J., Wang, J.: Domain-adapted feature transfer: A generalized framework for short-term vessel speed prediction. Ocean Engineering 280, 114536 (2023)

[31] Hong, H., Kim, W., Kim, S., Lee, K., Kim, S.S.: Deep transfer learning for efficient and accurate prediction of composite pressure vessel behaviors. Composites Part A: Applied Science and Manufacturing 186, 108413 (2024)

[32] Deng, Y., Li, Y., Zhu, H., Fan, S.: Displacement values calculation method for ship multi-support shafting based on transfer learning. Journal of Marine Science and Engineering 12(1), 36 (2023)

[33] Sharma, A., Altan, D., Marijan, D., Maressa, A.: From high-frequency sensors to noon reports: Using transfer learning for shaft power prediction in maritime. In: International Joint Conference on Computational Intelligence, pp. 723–741 (2025). Springer

Appendix A Robustness check

Validation of ALS_App predictions: We validate whether the models trained with ALS_App have learned the underlying patterns in weather and vessel operational data. In this evaluation setup, fused NRs are first averaged and then a single prediction is made. Earlier model predictions are made at 15-minute intervals and averaged afterwards to match the NR's daily average power value. Table A1 presents the evaluation metrics obtained using the revised evaluation paradigm for single-value prediction.

Table A1 Validation of aggregate loss supervision using single prediction evaluation strategy (S-1).

V	R^2	MAPE	MAE
A-1	0.66±0.1	17.70±3.4	708.1±145.2
A-2	-1.26±0.3	33.83±5.2	1882.9±265.6
A-3	0.65±0.0	17.95±1.7	531.3±26.5
A-4	0.79±0.4	12.08±9.0	557.4±303.9
A-5	0.40±0.1	14.31±1.5	679.5±42.1
A-6	0.33±0.2	21.01±2.8	1095.0±130.0
A-7	-1.82±3.1	11.92±4.2	613.9±223.0
A-8	-0.07±0.4	16.66±2.6	756.2±207.4

We compare the performance metrics from this evaluation with those in Table 3 for ALS_App. The results are comparable in terms of MAPE, with slight increases observed for A-4 and A-5, while the remaining vessels show consistent values across both tables. For R^2, most vessels show negligible changes. The model performance remains mostly the same across the two evaluation strategies, indicating that ALS_App generalizes well.

Reconstruction fidelity of the shared latent space: For the power prediction task, the quality of the shared latent space is critical, as a poorly structured feature representation can propagate errors into the downstream regression task. To validate the robustness of the latent space, we evaluate the reconstruction task independently. It provides evidence for encoder learning accurate patterns and allowing the decoder to reconstruct high-frequency data from the compressed latent code. Table A2 presents the reconstruction performance of overall and individual NRs features, encoded into the latent space and reconstructed at high frequency by the decoder. Performance is measured by evaluation metrics (R^2 and MAE) across vessels over the same testing period. We exclude MAPE from the evaluation because for most features, it becomes undefined due to very small denominator values, making the metric uninformative. The columns represent the vessels, features and metrics.

The results demonstrate that relevant features such as draft and wave height are reconstructed with high fidelity (R^2 as 0.92 and 0.86 respectively), confirming that the encoder learns the underlying input variance within the shared latent space. Whereas directional features such as relative wave and wind direction show substantially lower reconstruction accuracy.

Table A2 High-frequency reconstruction performance of features from the learned latent space using the encoder-decoder framework (S-1).

V	Features	R^2	MAE
A-1	Overall	0.65±0.01	0.86±0.03
	Speed	0.70±0.04	0.70±0.06
	Draft	0.88±0.02	0.53±0.05
	Wave height	0.77±0.02	0.30±0.01
	Rel. wave dir.	0.22±0.05	0.47±0.01
	Rel. wind dir.	0.66±0.02	2.29±0.08
A-2	Overall	0.67±0.01	0.91±0.03
	Speed	0.78±0.05	0.67±0.08
	Draft	0.89±0.02	0.48±0.03
	Wave height	0.77±0.01	0.36±0.01
	Rel. wave dir.	0.24±0.04	0.49±0.01
	Rel. wind dir.	0.66±0.02	2.53±0.07
A-3	Overall	0.60±0.01	0.84±0.02
	Speed	0.58±0.01	0.63±0.01
	Draft	0.80±0.02	0.62±0.03
	Wave height	0.83±0.01	0.29±0.01
	Rel. wave dir.	0.12±0.04	0.50±0.01
	Rel. wind dir.	0.69±0.01	2.14±0.06
A-4	Overall	0.47±0.02	1.05±0.03
	Speed	0.51±0.11	0.84±0.12
	Draft	0.75±0.02	0.84±0.03
	Wave height	0.71±0.03	0.32±0.01
	Rel. wave dir.	-0.05±0.04	0.55±0.01
	Rel. wind dir.	0.42±0.07	2.69±0.12
A-5	Overall	0.66±0.01	0.75±0.02
	Speed	0.33±0.03	0.83±0.02
	Draft	0.92±0.02	0.43±0.04
	Wave height	0.86±0.01	0.22±0.01
	Rel. wave dir.	0.41±0.04	0.40±0.02
	Rel. wind dir.	0.80±0.01	1.88±0.06
A-6	Overall	0.54±0.02	1.00±0.03

V	Features	R^2	MAE
	Speed	0.41±0.05	0.89±0.04
	Draft	0.78±0.03	0.70±0.06
	Wave height	0.69±0.02	0.38±0.01
	Rel. wave dir.	0.17±0.05	0.48±0.01
	Rel. wind dir.	0.63±0.02	2.54±0.10
A-7	Overall	0.64±0.01	0.82±0.02
	Speed	0.43±0.04	0.84±0.04
	Draft	0.88±0.01	0.64±0.04
	Wave height	0.75±0.02	0.30±0.01
	Rel. wave dir.	0.39±0.02	0.39±0.00
	Rel. wind dir.	0.74±0.01	1.93±0.04
A-8	Overall	0.70±0.01	0.75±0.02
	Speed	0.66±0.01	0.52±0.01
	Draft	0.88±0.02	0.53±0.06
	Wave height	0.81±0.01	0.26±0.01
	Rel. wave dir.	0.40±0.04	0.43±0.01
	Rel. wind dir.	0.76±0.01	2.01±0.04