Applied Ocean Research 136 (2023) 103583

Contents lists available at ScienceDirect

Applied Ocean Research

journal homepage: www.elsevier.com/locate/apor

Hybrid modelling to improve operational wave forecasts by combining process-based and machine learning models

Joost P. den Bieman*, Menno P. de Ridder, Marisol Irías Mata, Joana C.C. van Nieuwkoop

Deltares, Department of Coastal Structures and Waves, Boussinesqweg 1, 2629HV Delft, The Netherlands

ARTICLE INFO

Keywords:
Operational forecasting
Hybrid modelling
Machine learning
XGBoost
Spectral wave modelling
SWAN

ABSTRACT

Operational wave forecasting plays an important role in ensuring safe navigation and in the prediction of tidal windows for harbour approach channels. The underlying nearshore process-based wave models need to be accurate for a wide range of different conditions, from more common mild wave conditions to the occasional high energy (storm) conditions. In this work, an innovative hybrid modelling approach is proposed to improve the accuracy of operational wave forecasts. An operational wave model is combined with a machine learning model which is trained using wave measurements within the wave model domain. This hybrid modelling approach is applied to the Dutch North Sea, covering four major harbour approach channels.

The final hybrid operational wave model results in a significant average error decrease compared to just the process-based model, amounting to 21.7% for the wave energy density and 25.3% for the wave direction. The error reduction for the spectral wave parameters is even larger, with a 33.3% smaller error in spectral wave height and a 38.8% smaller error in spectral wave period. As this approach is generically applicable to spectral wave models, it contains the potential for significant improvements in operational modelling.

1. Introduction

In operational forecasting, nearshore process-based wave models are crucial in ensuring safe navigation and the prediction of tidal windows for harbour approach channels. For these applications, the nearshore wave model needs to be accurate for a wide range of different conditions, from more common mild wave conditions to the occasional high energy (storm) conditions.

Traditionally, there are several ways in which the accuracy of wave models is ensured. Firstly, wave models are both calibrated and validated based on field measurements, for instance from wave buoys. Such calibration and validation efforts are often either a one-time effort during the model setup or a periodical quality check of the operational model performance. The underlying assumption with this approach is that (part of) the discrepancy between model outcome and measurement can be solved by tuning the model settings. Another approach is to derive and implement new or improved descriptions of physical processes in the process-based model. This usually is a time and knowledge intensive activity, requiring rigorous research.

In the end, process-based wave models are very useful, but not perfect tools. Even when all the above-mentioned avenues for model improvement have been exhausted, in practice the operational wave model will still not give perfect predictions compared to the measurement data, because it is practically impossible to capture all physical processes in a process-based model. The input of a wave model is often a factor of large uncertainty. For this reason data assimilation is often used in forecasting, where measurement data are assimilated to improve the model input. However, in the practice of nearshore wave forecasting this technique is not often used, since data assimilation usually features a significant computational demand which is often infeasible in an operational context.

Another way to use a data-driven model is to make use of a machine learning (ML) approach. In many cases, there will be continuous wave measurements – such as wave buoys or measurement equipment on offshore platforms – available within the model domain. When both the predictions from the operational model and the measurements are stored for sufficiently long periods of time, these can provide the necessary training data for a ML model used to improve wave predictions. Existing efforts to incorporate ML models in operational forecasting can be divided into two categories: (a) those that completely replace the process-based wave model with a data-driven model and (b) those that use both a process-based and a data-driven model together. The former category contains examples of neural networks predicting of wave heights inside a harbour (Tsai et al., 2002) and wave heights in the

[*] Corresponding author.
E-mail addresses: joost.denbieman@deltares.nl (J.P. den Bieman), menno.deridder@deltares.nl (M.P. de Ridder), marisol.iriasmata@deltares.nl (M. Irías Mata), Joana.vanNieuwkoop@deltares.nl (J.C.C. van Nieuwkoop).

https://doi.org/10.1016/j.apor.2023.103583
Received 23 January 2023; Received in revised form 21 April 2023; Accepted 28 April 2023
Available online 11 May 2023

swell and low frequency bands (Lopez et al., 2015), both for a single location, and the use of Convolutional Neural Networks to predict waves and hydrodynamics as a surrogate for numerical models (Wei and Davison, 2022). For the latter category either Bayesian Networks (Emmanouil et al., 2020), deep learning models (Yevnin et al., 2023), Random Forests or Gradient Boosting Decision Trees (Callens et al., 2020) are used in conjunction with a process-based model to correct spectral wave parameters for a very small number of locations (one or two) or perform a short-term prediction of the wave height time series in the case of Yevnin et al. (2023). Also, de Ridder et al. (2021) propose to correct the remaining error between the process-based model result and the measurement on the entire wave spectrum instead of only on the spectral parameters (as in the previously mentioned works), which results in a much wider range of possible applications. This approach not only leads to significant improvements in the energy and mean directional spectra, but also in the spectral wave parameters derived from those corrected wave spectra. In this paper, this exploratory work is innovated upon by extensive optimization of the configuration of the ML model, firstly selecting and engineering the right input features to use and secondly deriving optimal settings. Furthermore it is expanded to accommodate a larger number of locations, a total of 14 along four major approach channels of harbours in the Netherlands, Belgium and Germany. For these locations, predictions are made by an operational SWAN wave model (Booij et al., 1996) of the Dutch North Sea. The ML correction is then predicted by a gradient boosting decision tree implementation – XGBoost (Chen and Guestrin, 2016) – in Python (van Rossum, 1995).

This paper is structured as follows. Section 2 describes the method and data, including the operational and machine learning models. In Section 3, the calibration of the machine learning model is discussed. The results of the calibrated model are shown in Section 4. The penultimate Section 5 contains the discussion and in Section 6 the conclusions drawn from this work are presented.

2. Method

The methods and data used within this work are described below, featuring in order: the operational wave model used (Section 2.1), the available data (Section 2.2), a description of the XGBoost ML model (Section 2.3) and finally the general approach and model setup (Section 2.4).

2.1. Operational wave model

To provide a wave forecast for the Dutch coast, the SWAN-Kuststrook model schematization (Gautier et al., 2018) is applied in combination with the third-generation wave model SWAN (Booij et al., 1996). The model is used by the operational forecasting service of the Dutch Directorate-General for Public Works and Water Management (Rijkswaterstaat), to provide a 48-hour forecast of the wave conditions. This forecast is updated every hours and the forecast data is available with a time step of one hour. The wave forecast provides essential data for early warning related to flooding and for operational shipping information, such as the computation of tidal windows for ships navigating the approach channels of the main Dutch ports.

The model domain covers the entire Dutch coast including the Wadden Sea and Western Scheldt (longitude 2.1° – 7.3°, latitude 51.1° – 54.1°). A spatially varying grid is applied with a grid resolution varying between 35 m nearshore and 6.2 km offshore. The directional space is divided into 45 bins, with a bin size of 8°. The frequency range is 0.03 – 0.6 Hz, divided into 32 bins. The offshore wave boundary is located at approximately 100 km out of the Dutch coast (see Fig. 1) and the ECMWF-WAM wave data (The Wamdi Group, 1988) is used as forcing. In addition to the wave forcing on the offshore boundary, the SWAN-Kuststrook model is forced with wind fields from the numerical weather prediction model HARMONIE (Bengtsson et al., 2017).

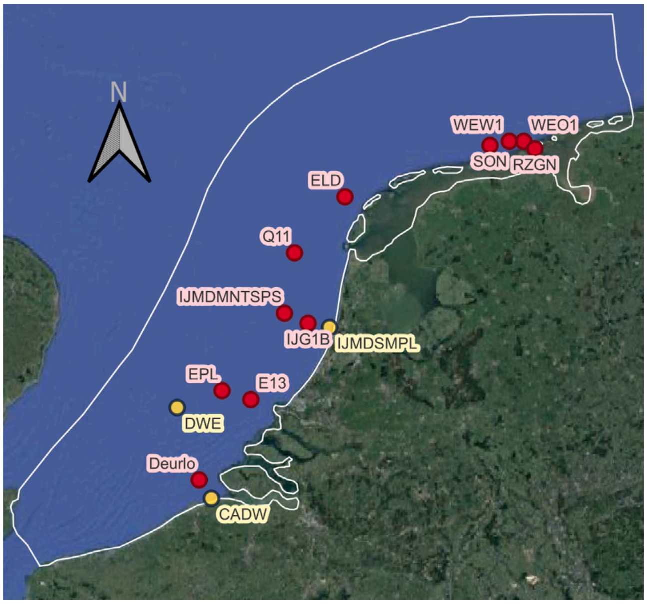


Fig. 1. Wave measurement locations (dots) and SWAN-Kuststrook model domain extent (white line). In red the locations for which 1.5D measured wave spectra are available (wave energy data and mean wave direction and directional spreading per frequency bin). In yellow the locations for which only 1D measured wave spectra are available (only wave energy data).

2.2. Available measurement data

To train the ML model, 14 wave measurement locations are selected from four regions, all of them close to the approach channels of major Dutch (and other European) harbours. Firstly, the region Wadden Sea (WAD) covers the approach to the ports of Eemshaven, Delfzijl and Emden (Germany). The second region is the IJgeul (IJG), the access channel to the Port of Amsterdam. The third region features the Eurogeul (EUG), the access channel to the Port of Rotterdam. The final region covers the mouth of the Western Scheldt (WES) estuary, which gives access to the ports of Vlissingen, Terneuzen and Antwerp (Belgium). The selected stations within these regions are shown in Fig. 1. Both wave measurements and output data from the SWAN-Kuststrook model are available for all 14 locations, as listed in Table 1.

The wave observation data consists of wave energy density and mean wave direction per frequency bin and per time step. This is called a 1.5D wave spectrum, since the distribution of wave energy over the frequency space is known, but the distribution of wave energy over wave directions is simplified by specifying a mean wave direction per frequency bin. A full 2D wave spectrum would contain the wave energy distribution over both frequency and directional space. The available wave observation data is available with a time step of 10 minutes and covers a period from October 2020 to April 2022 (about 18 months) for the directional wave information. For wave energy density, the data is available up to 18 January 2022 (about 15 months). Incidentally, there are gaps in the time series, but in general the measurement data set is fairly complete. Whether or not wave directions are measured is dependent on the type of measuring device deployed at a location. At the locations Eurogeul DWE, IJmuiden stroommeetpaal and Cadzand CADW no recordings of wave direction or directional spreading are available (these locations are indicated by * in Table 1).

The wave energy, wave direction, water level, significant wave height (H_{m0}) and spectral period ($T_{m-1.0}$) are extracted from the SWAN-Kuststrook model through the MATROOS web tool (Rijkswaterstaat, 2022).

Since the HARMONIE wind speed and wind direction is only available for a small number of locations that do not overlap entirely with

Table 1
Overview of the locations and the number of predicted wave spectra for each location. The measurements that lack information on wave direction are indicated by *.

Location name	Location ID	N	Wind location	Region
Randzelgat Noord	RZGN	3,730	Huibertgat	WAD
Westereems Oost	WEO1	11,148	Huibertgat	WAD
Westereems West	WEW1	8,368	Huibertgat	WAD
Schiermonnikoog Noord	SON	10,501	Huibertgat	WAD
Eierlandse Gat	ELD	9,389	Q11	WAD
Platform Hoorn Q11	Q11	9,413	Q11	WAD
IJmuiden munitiestortplaats	IJMDMNTSPS	10,150	IJMDMNTSPS	IJG
IJgeul 1 boei	IJG1B	11,069	IJMDMNTSPS	IJG
IJmuiden stroommeetpaal*	IJMDSMPL	10,498	IJMDMNTSPS	IJG
Europlatform	EPL	11,215	EPL	EUG
Eurogeul E13	E13	10,670	EPL	EUG
Eurogeul DWE*	DWE	10,039	EPL	EUG
Deurlo	Deurlo	10,933	Vlakte vd Raan	WES
Cadzand CADW*	CADW	6,238	Vlakte vd Raan	WES
All locations	–	133,361	–	–

Table 2
Number of available data points (number of wave spectra for the locations within the region multiplied by the number of frequency bins) for the different regions. Both the total, training, validation and test set are shown.

Region	Total	Training	Validation	Test
Wave energy				
WAD	1,471,372	1,064,021	217,933	189,418
IJG	888,076	626,295	128,279	133,502
EUG	893,872	642,494	131,595	116,783
WES	480,788	338,514	69,336	72,938
Wave direction				
WAD	1,695,036	1,248,872	255,792	190,372
IJG	666,400	470,843	96,437	99,120
EUG	646,156	474,027	97,089	190,372
WES	306,600	216,318	44,306	45,976

the all wave measurement locations, 5 reference wind locations have been chosen that provide wind information for a certain area and coupled to the wave measurement locations (see Table 1). The SWAN prediction, wave measurements and wind input are synchronized along the time axis, resulting in a data set with a communal time step of one hour.

Once all the information from the different data sets is gathered, the wave measurement data, SWAN data and wind data combined into one large data set per measurement location. The number of data points per region is listed in Table 2. Fig. 2 gives an illustration of the final data set for all the locations by showing the density scatter plots of the wave height, wave period, wave direction, wind speed and wind direction. These scatter plots show that the data set contains different hydrodynamic conditions. Both mild ($H_{m0} < 1m$) and more severe ($H_{m0} > 3m$) wave conditions are included in the data set.

2.3. Machine learning model

In this work, the Python (van Rossum, 1995) package implementing the machine learning method XGBoost (Chen and Guestrin, 2016) is applied. Existing examples of the successful application of XGBoost in adjacent fields include the aforementioned wave parameter correction (Callens et al., 2020), the prediction of added-wave resistance on ships (Mittendorf et al., 2022) and the prediction of mean wave overtopping volumes (den Bieman et al., 2021).

XGBoost falls under the category of models named gradient boosting decision trees (GBDT), which make use of decision trees. These models can either predict a label (classification) or a quantity (regression). The term used for this type of decision trees is CART (classification and regression trees).

The anatomy of a CART consists of decision and leaf nodes. Decision nodes contain a condition based on a training data feature, also known as a split. An example of a split could be “Is the wave height larger than 1.5 m?”. From the split, two branches emerge corresponding to the two answers. Each branch in turn ends in a next decision or leaf node. A leaf node gives a prediction of the target variable, so no further branches stem from a leaf node. A CART starts at a single decision node and branches out from there, with branches eventually ending in leaf nodes. The maximum number of decision nodes between the start and a leaf node is called the depth of the CART.

Usually, the complexity of real-world problems is too large to solve with a single CART. This is why GBDT models (such as XGBoost) make use of an ensemble of CARTs containing a (very) large number of them. The idea behind that is that, while a single CART is a weak predictor, the sum of many of them can result in a strong predictor. In the ensemble, the predictions of each individual CART are summed up to the ensemble prediction, accounting for the learning rate (see Section 3.4) applied in the model.

In the training of a GBDT model, new trees are iteratively added, each one trying to reduce the remaining prediction errors of the trees already present in the ensemble. When forming a new tree, its splits must be determined. To that end, an objective function is specified. The objective function consists of two parts, a training loss function that promotes prediction accuracy of the model on the training data set and a regularization term that penalizes the complexity of the tree. Additional levels of tree depth are iteratively added, searching for the optimal split condition per feature. Then the combination of feature and split condition that give the largest improvement in terms of the objective function is selected for the decision node. The maximum number of levels that is grown in this way is capped by the maximum tree depth. The work by Chen and Guestrin (2016) extensively details the entire XGBoost algorithm.

The total dataset for each region is randomly divided into training, validation and test data sets (see Table 2). The model is trained on the training data set, during which the validation data set is used in the early stopping algorithm described below. To show the performance of the model it is applied to the up to now unseen test data set, the results of which are presented in Section 4. The split is approximately 70% training, 15% validation and 15% test data. Some deviations from this split occur, because the test data set covers the same moments in time for all locations but sporadically data is missing for some of those locations. Note that each moment in time is exclusively used within either the training, validation or test data set. Since the model only uses the input of a single moment in time (as is described in Section 2.4), there is no data leakage between these data sets.

The number of CARTs that form the ensemble can either be a hard number set as input variable or determined during the training process,

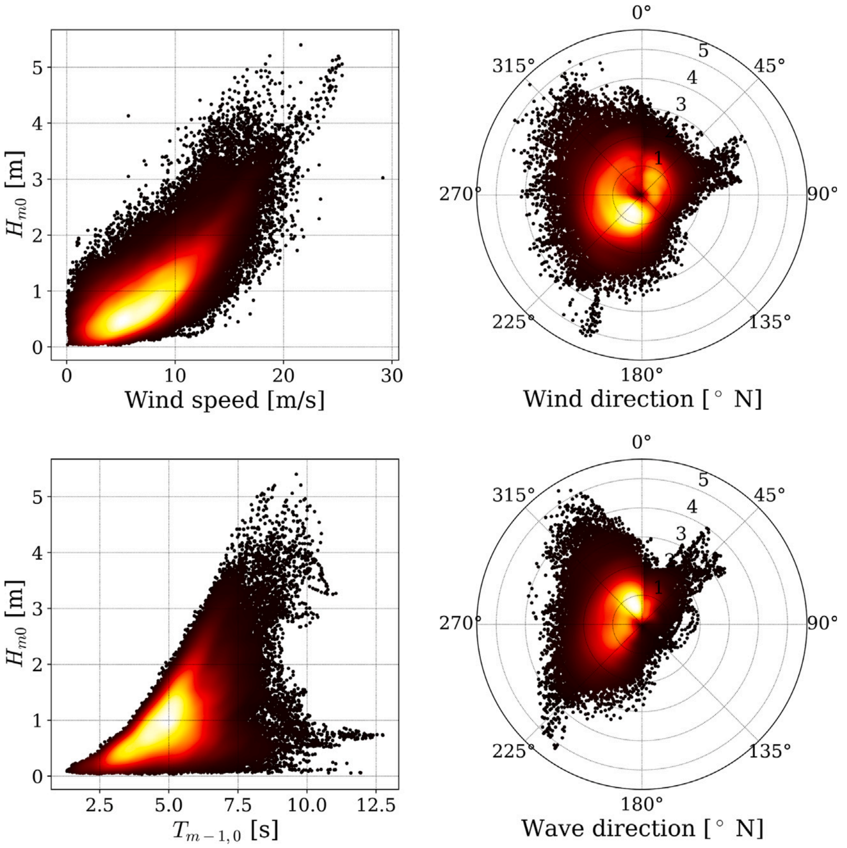


Fig. 2. SWAN wave data and HARMONIE wind information for all locations listed in Table 1. The upper left panel shows the density scatter of the wave height and the wind speed. The upper right panel shows the density scatter of the wave height and the wind direction. The lower left panel shows the density scatter of the wave height and the spectral period. The lower right panel shows the density scatter of the wave height and the wave direction. This data set contains 8.368 predicted wave spectra.

using a method called ‘early stopping’. Early stopping stops adding trees when the model improvement from new trees becomes too small. The algorithm adds new trees until another new tree would either exceed the maximum number of trees or the last 1,000 trees no longer improve the models performance on the validation data set.

2.4. Approach and model setup

To improve operational wave predictions, a ML model (Section 2.3) is trained to predict the correction to the SWAN-Kuststrook model (Section 2.1). The ML model is trained using the SWAN model predictions combined with wind input and wave measurement data (Section 2.2). Here, the wave measurement data is used to derive the target variable, namely the correction necessary to make the SWAN output match the wave measurements. Corrections are derived for each bin of the frequency spectrum, both for the energy density and the mean direction. Since the energy density and the mean direction are two separate target variables this requires two ML models, one energy model and one direction model.

The ML model has the following SWAN related variables available (also listed in Table 3): the HARMONIE wind velocity (U_{wind}) and direction (D_{wind}) at the associated wind location (see Table 1), the water level (h), spectral wave height (H_{m0}), spectral wave period ($T_{m-1.0}$) and mean wave direction (θ), the energy density ($E(f)$) and mean wave direction ($\theta(f)$) per frequency bin at a specific location. In addition to these SWAN related variables, a set of input features also includes indicators to both the location and the bin in the frequency spectrum for which the correction is being predicted. This means that a single wave spectrum at one location translates to 28 data points, since the frequency spectrum is discretized into 28 bins. So for each of these 28 frequency bins the variables listed in Table 3 are provided as input and the only variables that change between frequency bins within the same wave spectrum are $E(f)$ and $\theta(f)$.

Table 3
Overview of input features for the machine learning model. The features excluded from the final model are indicated by *.

Feature name	Symbol	Unit
HARMONIE wind velocity used in SWAN	U_{wind}	[m/s]
HARMONIE wind direction used in SWAN	D_{wind}	[°N]
Water level used in SWAN	h	[m +NAP]
Spectral wave height predicted by SWAN	H_{m0}	[m]
Spectral wave period predicted by SWAN*	$T_{m-1.0}$	[s]
Deep water wave steepness predicted by SWAN	$s_{m-1.0}$	[–]
Wave direction predicted by SWAN*	θ	[°N]
Difference between wave and wind direction	$\delta\theta$	[°N]
Predicted energy density in frequency bin	$E(f)$	[m^2/Hz]
Predicted main wave direction in frequency bin	$\theta(f)$	[°N]
Location indicator	–	[–]
Frequency bin indicator	–	[–]

As mentioned in Section 2.2 the wave measurements are only available as 1.5D spectra. To match the format of the wave measurements, the 1.5D spectra from the operational SWAN model are used and interpolated to match the discretization of the measured wave spectra. In this way, both measured and modelled spectra can be added to the same training data base. Note that, if there were 2D measured wave spectra available, the method could still be applied when expanded with directional bins.

Both the different locations and frequency bins are represented using ‘one-hot-encoding’. This means that each different location and frequency bin is added as a feature (column) in the training data. Each row in the data contains a value of 1 for the feature representing the location and frequency bin it pertains to, with all other locations and frequency bins containing a value of 0. In this way, the model is able to differentiate between the different locations and frequency bins. Applying one-hot-encoding to the frequency bins adds 28 features, for the locations it adds between 2 and 6 features depending on the number of locations in a region (see Table 1).

For the energy model the target variable is a normalized correction of the energy density in a given frequency bin ($\Delta E(f)$). It is defined as the difference between the observed ($E_{obs}(f)$) and SWAN prediction of the energy density in that bin ($E(f)$), which is then normalized by the total amount of predicted energy ($E_{SWAN,total}$) over all frequency bins (see Eq. (1)). The normalization step effectively reduces the dimensionality of the problem, because both mild and more extreme wave conditions result in a similar range of values for this target variable. In other words, if the normalization is left out, the range of values of the target variable becomes much wider, making it harder for the ML model to accurately predict. Note that the algorithm is prevented from predicting negative energy density, in which case it is set to 0.

$$\Delta E(f) = \frac{E_{obs}(f) - E(f)}{E_{SWAN,total}} \tag{1}$$

For the direction model the target variable is the normalized correction of the main wave direction in a given frequency bin ($\Delta\theta(f)$). It is defined as the difference between the observed ($\theta_{obs}(f)$) and the SWAN prediction of the main wave direction in that bin ($\theta(f)$), normalized by 360° (see Eq. (2)). Note that the difference between the two directions has a value between −180° and +180°, since the smallest directional difference is used.

$$\Delta\theta(f) = \frac{\theta_{obs}(f) - \theta(f)}{360} \tag{2}$$

The root-mean-squared error (RMSE) – defined in the Appendix – is used both as the objective function in the ML model training and to evaluate the performance of the SWAN and corrected SWAN results.

Since the SWAN-Kuststrook model is an operational model, its predictions are available with different lead times. To train the ML model, the SWAN prediction with zero lead time – the 'nowcast' – is used. The main consideration here is that for a given moment in time, the predictions for that moment with different lead times mainly feature different levels of uncertainty pertaining to the SWAN model forcing, i.e. the wind conditions and water levels. The aim of this work is to train a ML model to correct for the error made by SWAN in the wave prediction, and not to correct for the errors in the wind and water level predictions as well. Effectively, the latter would be the case when using predictions with a lead time. Even though the model is trained on the nowcast, it is suitable for application to SWAN predictions with any amount of lead time, the model just will not correct for any prediction errors in the SWAN model forcing. Note that a somewhat similar effort is of course possible to create another ML method that does correct the predicted wind and water level, as showcased by the work of Yevnin and Toledo (2022). This however is out of scope of the current work.

3. Model calibration

To arrive at the final ML model used in this work the model is calibrated in a few ways. Firstly, the question of how to group the 14 different locations is addressed in Section 3.1. Then in Section 3.2 several features in the training data set are recast in an attempt to make the model more efficient. Subsequently, both the importance of the different training features (Section 3.3) and the optimal model settings (Section 3.4) are analysed.

3.1. Location grouping

As mentioned in Section 2.2, there are 14 locations of interest in the operational wave model domain which are included in the ML model. Including all locations in a single model will significantly increase the computational effort of training the model, especially since each location adds a column to the input data due to the one-hot-encoding described in Section 2.4. Because of the large extent of the operational wave model domain some of the locations considered in this work are relatively far apart, so that in practice they are located in water systems that behave differently. For instance the locations in the south along the mouth of the Western Scheldt are so different from those in the north close to the Wadden Sea, it is questionable whether it would even be advantageous to include both in the same ML model. Hence, both from a physics and a practical standpoint, the locations are grouped into four regions as described in Section 2.2. In this way, four different ML models are trained with the same model setup.

Table 4
Root-mean-squared error (RMSE) [m^2/Hz] for three different training sets (EUG+IJG, EUG and location DWE). The RMSE is shown for all three locations within region EUG, both for the uncorrected energy density (SWAN) and the corrected energy (SWAN$_{corr}$).

Loc.	SWAN	EUG+IJG SWAN$_{corr}$	EUG SWAN$_{corr}$	DWE SWAN$_{corr}$
E13	0.38	0.29	0.30	0.32
EPL	0.40	0.31	0.31	0.33
DWE	0.43	0.36	0.36	0.37

To verify how the selection of these groups affects the final accuracy, ML have been trained with three different location subset variations in the training data set: (1) a training set with only location Eurogeul DWE, (2) all locations in the Eurogeul region, and (3) a combination of the locations from both Eurogeul and IJgeul regions (see Table 1). Note that the associated test data sets have the same moments in time for each location (if available) in order to fairly compare the results.

The results, listed in Table 4, indicate that the model is fairly robust for the grouping of locations in the training data set, and different variations thereof do not significantly affect the model performance. All three variations show a similar accuracy for the corrected energy density. This indicates that an enlarged training data set with additional (nearby) locations does not improve the accuracy of the ML model. Thus, from a practical standpoint it is decided to set up ML models for the four regions (WAD, IJG, EUG and WES) separately. This seems a sensible choice as it both prevents a large number of different models on the one hand and prevents a large amount of one-hot-encoded location features in the training data set on the other. To show whether a correction is also applicable for nearby locations, the Eurogeul DWE model trained on only location DWE is also used to predict the correction for the locations Eurogeul E13 and Eurogeul EPL. These suggest that in this case the correction is also applicable for nearby locations, since the error reduction for E13 and EPL is fairly similar to (though slightly worse than) the error reduction of the location DWE.

3.2. Feature engineering

In an effort to increase the accuracy of the ML model, the input features are adapted to make it easier for the algorithm to discover the patterns present in the data. One strategy to do this is to redefine input features to make them uncorrelated to other features whenever possible. In this work, that can easily be done with both $T_{m-1.0}$ and θ. The spectral wave period $T_{m-1.0}$ is partially correlated mainly to the H_{m0}, in the sense that higher wave heights will also results in longer wave periods. By using the deep water wave steepness, $s_{m-1.0}$ (see Eq. (3)), as a feature instead of the wave period this problem is avoided, since H_{m0} and $s_{m-1.0}$ are uncorrelated.

$$s_{m-1.0} = \frac{2\pi \cdot H_{m0}}{g \cdot T_{m-1.0}^2} \tag{3}$$

In a similar vein, the wave direction is often partially correlated to the wind direction. To avoid this correlation in the training features, the wave direction is replaced with $\delta\theta$ which represents the difference

Table 5
Overview of feature importance rankings for the different regional models for the energy and mean direction in the wave spectrum. Note that the one-hot-encoded features related to location and frequency bin are not part of the feature importance analysis.

Rank	WAD	IJG	EUG	WES
Wave energy				
1	H_{m0}	H_{m0}	H_{m0}	H_{m0}
2	$s_{m-1.0}$	$s_{m-1.0}$	$s_{m-1.0}$	$s_{m-1.0}$
3	$E(f)$	$E(f)$	$\delta\theta$	$E(f)$
4	U_{wind}	$\delta\theta$	D_{wind}	U_{wind}
5	$\delta\theta$	U_{wind}	U_{wind}	h
6	D_{wind}	h	h	D_{wind}
7	h	D_{wind}	$E(f)$	$\delta\theta$
Wave direction				
1	$\theta(f)$	$\theta(f)$	$\theta(f)$	$\theta(f)$
2	D_{wind}	D_{wind}	D_{wind}	D_{wind}
3	$\delta\theta$	H_{m0}	H_{m0}	H_{m0}
4	H_{m0}	$\delta\theta$	$\delta\theta$	$\delta\theta$
5	U_{wind}	$s_{m-1.0}$	$s_{m-1.0}$	$s_{m-1.0}$
6	$s_{m-1.0}$	U_{wind}	U_{wind}	U_{wind}
7	h	h	h	h

between wind and wave direction (see Eq. (4)). The possible values of $\delta\theta$ range from −180° to +180°. Note that this only replaces the wave direction for the spectrum as a whole. The predicted main wave direction per frequency bin, $\theta(f)$, is still included as a feature in the directional model.

$$\delta\theta = \theta - D_{wind} \tag{4}$$

To see what effect these changes have the directional model of the WES region is trained with both the original and the new uncorrelated features. The effect on the accuracy of the model is negligible, but the model with the uncorrelated features converges much faster, with 150 trees required in the XGBoost model instead of 237 trees using the original features. Apparently, the patterns present in the data are now more easily found by the model, making it converge faster and providing a significant reduction in the model training effort. Hence, the new uncorrelated features are adopted going forward.

3.3. Feature importance

When making a final selection of which features should comprise the input for the ML model and which should be left out, a feature importance analysis can be a useful tool. The idea is that this analysis results in a ranked list of features from most to least important for the model outcome of the chosen target variable (in this case either the amount of energy or main direction in a given frequency bin). To arrive at this ranking, the feature importance is derived by performing a permutation importance analysis (Breiman, 2001; Fisher et al., 2018). This method iterates over the available features, and one-by-one scrambles a single feature in the test data set to gauge its effect on predictions of the target variable, with important features having a large effect and vice versa. To account for randomness, the scrambling of a feature is repeated 5 times and the importance is averaged for each feature. In this work, the permutation importance implementation in the ELI5 (2020) Python package was used. Note that the one-hot-encoded features related to location and frequency bin are not considered in the analysis, since permutation of one of those features would lead to input combinations that belong to two frequency bins or locations at the same time, which is nonsensical.

From Table 5 it can be seen that for the energy models, the H_{m0} and $s_{m-1.0}$ are consistently the two most important features, representing 56.1% and 29.0% of the average total weight respectively. Their feature importance is significantly higher than that of the features ranked 3–7. This makes sense, since these two variables form a basic characterization of the spectrum concerning the total amount of energy (H_{m0}) and the distribution of the energy over the frequency bands ($s_{m-1.0}$). The remaining features for the energy models are all much closer together in terms of importance (ranging from 1.0% to 7.8%), so depending on the specific region the specific ranking of these features varies a bit.

Table 6
XGBoost parameter variations used in the hyperparameter tuning process, with optimal values indicated by *.

Name	Parameter name	Values
Max. tree depth	*max_depth*	15; 25*
Min. data points per leaf	*min_child_weight*	50*; 100
Learning rate	*learning_rate*	0.05*; 0.10

Similar to the energy models, the two highest ranked features for the directional models - $\theta(f)$ and D_{wind} - are consistently the most important with some distance from the rest, accounting for at 45.4% and 34.2% of the average total weight. Logically, the wave direction in the given frequency bin predicted by SWAN ($\theta(f)$) and the HARMONIE wind direction (D_{wind}) should already provide quite a lot of information regarding the correct prediction of the direction. Again, the features at ranks 3–7 are fairly close together in their importance (ranging from 2.0% to 6.3%), with the total amount of energy (H_{m0}) and the difference between wind and wave direction ($\delta\theta$) being the most important among them.

Overall, it can be said that the most important features for the ML model seem to make sense from a physical standpoint. Additionally, as expected they are fairly consistent over the different regions, which adds to the confidence in the ML models.

3.4. Hyperparameter tuning

The process of finding the optimal parameter settings for a specific application of a ML method is often called ‘hyperparameter tuning’. The optimal settings form a compromise between unrestricted model complexity, which promotes model overfitting, and a model that is too simple, rendering it unable to represent the intricacies of the underlying data.

Three hyperparameters are tuned: the maximum tree depth (*max_depth*), the minimum number of data points per leaf (*min_child_weight*), and the learning rate (*learning_rate*). The maximum tree depth limits complexity of a singular decision tree, where the depth is the number of subsequent decision nodes in the tree. Using a minimum number of data points per leaf prevents trees becoming very specific – their structure being determined by a single point in the training data set – at the cost of their generic predictive skill. Lastly, the learning rate is implemented as a multiplication factor on the prediction of each tree in the ensemble, where values < 1 result in slower but often more stable conversion to a final trained model.

To find these optimal settings, a K-fold cross-validation (with $K = 3$) is used in combination with a grid search. In the grid search, a grid of values for the hyperparameters that are being optimized is constructed. For every combination of hyperparameter values, a model is trained and its performance assessed on a test data set. The 3-fold cross-validation repeats each model training three times with a different ‘fold’ – the random split between training and test data – to lessen the influence of this random split on the optimal settings that are found.

The values in the hyperparameter optimization are shown in Table 6. The tuning is executed for each region for both energy and direction models. The large majority of the results points in the direction of the following optimal set of hyperparameters: a maximum tree depth of 25, a minimum of 50 data points in the leaf nodes and a learning rate of 0.05. In all cases, the differences in performance for the different sets of hyperparameters was very small, indicating that – at least for this value grid – any combination of parameter values could lead to an accurate ML model.

Table 7
Overview of root-mean-squared errors of energy and directional wave spectra for the test data set per region, including the relative error reduction (in brackets).

	Energy	[m^2/Hz]	Direction	[°]
	SWAN	SWAN$_{corr}$	SWAN	SWAN$_{corr}$
WAD	0.52	0.40 (−23.1%)	50.3	37.3 (−25.8%)
IJG	0.51	0.40 (−21.6%)	48.2	34.4 (−28.6%)
EUG	0.40	0.32 (−20.0%)	53.2	42.6 (−20.0%)
WES	0.20	0.18 (−10.0%)	50.4	37.0 (−26.5%)
All regions	0.46	0.36 (−21.7%)	50.4	37.6 (−25.3%)

4. Results

In this section, the calibrated XGBoost ML model is applied to the test data set to gauge its performance. Firstly, this performance is quantified in Section 4.1 using the target variables for the correction of wave energy density and direction. Then, in an effort to translate this into practical terms, in Section 4.2 use is made of spectral wave parameters to express the model performance.

4.1. Target variables

Based on the predicted correction and the initial SWAN spectrum, a 1.5D wave spectrum for the test data set (previously unseen by the ML model) was reconstructed. To illustrate the performance of the ML model for all the data points in the test data set (see Section 2.3), a density scatter plot is shown in Fig. 3 for the energy density and the mean wave direction. The density scatter for the directions is shown for the absolute direction error bounded between −180° and +180°. It can be seen that the energy density scatter clouds of the corrected spectrum (panel b) contain less scatter and are better aligned with the line of perfect agreement compared to the original spectrum (panel a). The mean wave direction also shows a scatter reduction when the corrected results (panel d) are compared to the original SWAN results (panel c). Note that for the direction there is a distinct high density area (lighter colours) as most of the points have an observed wave direction between 300 and 360°N. Applying the correction moves this high density area closer to the zero-error line, which means that the wave direction for waves from this directional sector is better represented in SWAN$_{corr}$. It is interesting that the scatter plots for the uncorrected SWAN directions (panel c) include clusters of data points around diagonal lines (from upper left to lower right). These clusters correspond to frequency bins of the lower frequencies for which SWAN has a preferred direction, whereas the observations contain all directions. This model artefact in SWAN results in a significant deviation in the direction, but it does not have a large effect on the integral wave parameters because these low-frequency bins do not contain much energy.

In Table 7 the root-mean-squared errors corresponding to the data points shown in Fig. 3 are listed for the SWAN model with ML correction (indicated by SWAN$_{corr}$) and without ML correction for both the energy and direction models. The statistical comparison shows that the SWAN$_{corr}$ model reduces the RMSE by more than 20% from 0.46 m^2/Hz to 0.36 m^2/Hz for the energy density and by more than 25% from 50.4° to 37.6° for the mean wave direction.

When comparing the different regions with each other, the initial errors in the wave direction are of roughly the same magnitude (around 50°) but apparently the corrections in IJG are easier to predict (28.6% improvement) than those in EUG (20.0% improvement). In terms of energy density it is remarkable that the initial error in the WES area is quite small compared to the other areas. This is mainly caused by the fact that the WES locations are located in relatively shallow water, resulting in lower wave energy. As is to be expected, the subsequent improvement of the SWAN$_{corr}$ model is relatively small compared to the other regions (10% error reduction).

Above, SWAN$_{corr}$ is shown to perform well for individual frequency bins. This, however, does not necessarily guarantee that it leads to realistic reconstructed 1.5D spectra. In Fig. 4 three examples are shown of 1.5D wave spectra. These three wave spectra include one mild and two more energetic wave conditions with wave heights exceeding 1.5 m. For examples the SWAN$_{corr}$ wave spectrum matches better with the observed wave spectrum. Both the shape (represented by the spectral period) and the total amount of energy (represented by the wave height) are more accurately captured. The SWAN$_{corr}$ mean wave direction also matches better with the observation than the uncorrected spectrum.

Table 8
Overview of root-mean-squared errors of spectral wave height and spectral wave period per region for the test data set.

	H_{m0} [m]	SWAN$_{corr}$	$T_{m-1.0}$ [s]	SWAN$_{corr}$
	SWAN		SWAN	
WAD	0.24	0.15 (−37.5%)	0.55	0.42 (−23.6%)
IJG	0.23	0.14 (−39.1%)	0.68	0.44 (−35.3%)
EUG	0.23	0.14 (−39.1%)	0.51	0.30 (−41.2%)
WES	0.15	0.12 (−20.0%)	0.83	0.43 (−48.2%)
All regions	0.21	0.14 (−33.3%)	0.67	0.41 (−38.8%)
$H_{m0} > 2m$	0.42	0.24 (−42.9%)	0.38	0.29 (−23.7%)
$H_{m0} \leq 2m$	0.16	0.11 (−31.3%)	0.70	0.42 (−40.0%)

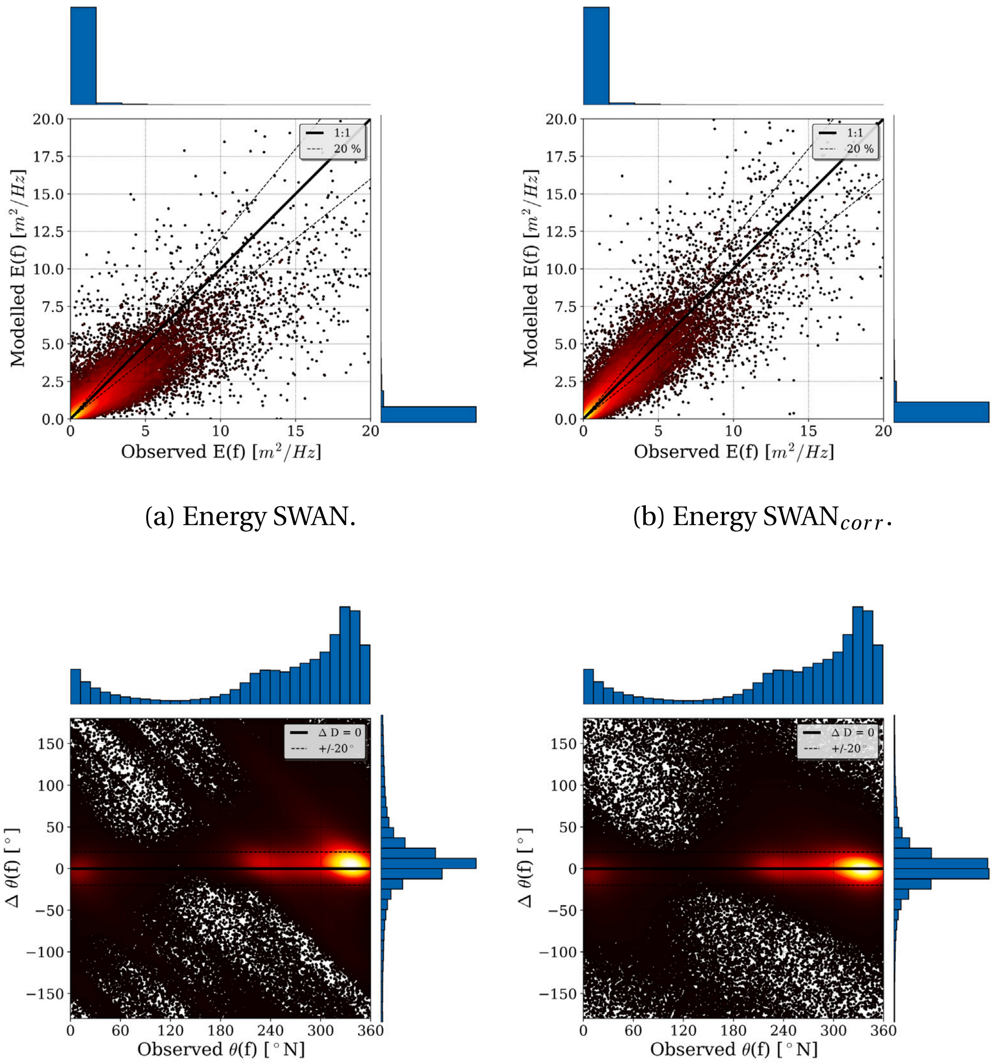
Overall, in terms of the target variables the SWAN$_{corr}$ results are a significant improvement over the SWAN results for both the wave energy density and direction.

4.2. Spectral wave parameters

In addition to the performance of SWAN$_{corr}$ on the energy density in individual frequency bins (Section 4.1), it is also insightful to asses its performance on integral spectral wave parameters, using the spectral wave height (H_{m0}) and the spectral wave period ($T_{m-1.0}$). The spectral parameters modelled by both SWAN and SWAN$_{corr}$ for the test data set are plotted against the observed values from the wave measurements in Fig. 5.

In Fig. 5(a) it can be seen that the SWAN$_{corr}$ successfully corrects a slight underestimation of larger wave heights, while also reducing the scatter. For the $T_{m-1.0}$, Fig. 5(b) shows that the main benefit of SWAN$_{corr}$ is that it significantly reduces over- and underestimation in the 3–6 s (observed) period range.

The impression of improved performance given by Fig. 5 is substantiated by Table 8, which lists the root-mean-squared error for H_{m0} and $T_{m-1.0}$ for both the initial SWAN prediction and the corrected prediction (SWAN$_{corr}$). In general, SWAN$_{corr}$ offers a large improvement of the spectral wave parameters, with the H_{m0} error decreasing by a third and the $T_{m-1.0}$ error decreasing with 38.8%. The initial errors and effectiveness of correction of the H_{m0} is very similar for three of the four regions. In line with the observations in Section 4.1 the initial H_{m0} error in the WES region is relatively small, which again leads to a smaller relative improvement (20%) by SWAN$_{corr}$, compared to the other regions (almost 40%). Conversely, the $T_{m-1.0}$ in the WES region is relatively poorly predicted by SWAN, but can effectively be corrected by SWAN$_{corr}$ to reduce the error with 48.2%. When the test data set is divided into a set with milder ($H_{m0} \leq 2m$) and more extreme wave conditions ($H_{m0} > 2m$), it can be concluded that for both sets the improvement is significant. Furthermore, this distinction shows that the uncorrected SWAN model has a larger spectral period RMSE for milder waves, which seems counter-intuitive. Since the SWAN model does not always accurately simulate the spectral shape for swell conditions (long waves with smaller wave heights) and very small waves, this results in a larger $T_{m-1.0}$ error for mild wave conditions.



(a) Energy SWAN.

(b) Energy SWAN$_{corr}$.

(c) Direction SWAN.

(d) Direction SWAN$_{corr}$.

Fig. 3. Density scatter plot of the energy density (panels a and b) and mean wave direction (panels c and d) for the original spectrum, SWAN, (left) and corrected spectrum, SWAN$_{corr}$, (right) for the test data set. The 20% error bands are shown for the energy density plots and the absolute deviation of 20° is shown for the mean wave directions. The directional error shown in panels c and d is computed as the observed direction subtracted from the modelled direction. The colours indicate the density of the point (number of instances), with yellow colours representing high density. A log colour scale is used for point density in the energy density figures, to make the interpretation of these panels easier. The histograms on the top and side of the panels show the density of the points of both the x-axis and y-axis.

5. Discussion

In this paper, a data-driven approach for correcting an operational wave model is presented. This approach suits the high complexity and dimensionality of both the data and its underlying physical processes. This becomes more than clear when comparing the performance of the ML model with that of a simple bias correction, which is defined as the mean normalized correction (mean of $\Delta E(f)$ or $\Delta\theta(f)$). Only correcting for this mean bias in wave energy density and direction would lead to error decreases of 2.2% and 0% respectively, compared to the 21.7% and 25.3% error reductions reached with the XGBoost model.

When comparing this work to the approaches described in existing literature to use data-driven techniques to improve operational modelling (see Section 1), two important differences are apparent: (a) in this work the entire wave spectrum is corrected instead of correcting just the spectral wave parameters and (b) a significantly larger domain with more locations is considered (14 instead of 1 or 2). Outputting a corrected wave spectrum opens up a whole range of new applications, as the wave spectrum is needed for instance to predict wave-ship interaction and consequently tidal windows for ports. At the same time, Section 4.2 show that correcting the wave spectrum still results in a significantly more accurate prediction of the spectral wave parameters. The larger number of locations enables this approach to cover all relevant locations along 4 major harbour approach channels in the North Sea, which can contribute to safe navigation along important stretches of shipping routes instead of just a single location.

The limitations of the data-driven model presented here can be separated into two categories: those that are inherent to the type of method (GBDT) and those that stem from the amount and quality of the data used in model training. Regarding the former, because of the tree structure of the CARTs, GBDT methods are generally not well equipped for extrapolation past the feature ranges of the data set they are trained on. In a similar vein, for regions in the feature space that are relatively sparsely populated with training data the tree structure will often be quite coarse, lacking the granularity for accurate predictions. Lastly, it can be easy to overfit regression trees on the training data. This can, amongst other options, be avoided by requiring leaf nodes to have a

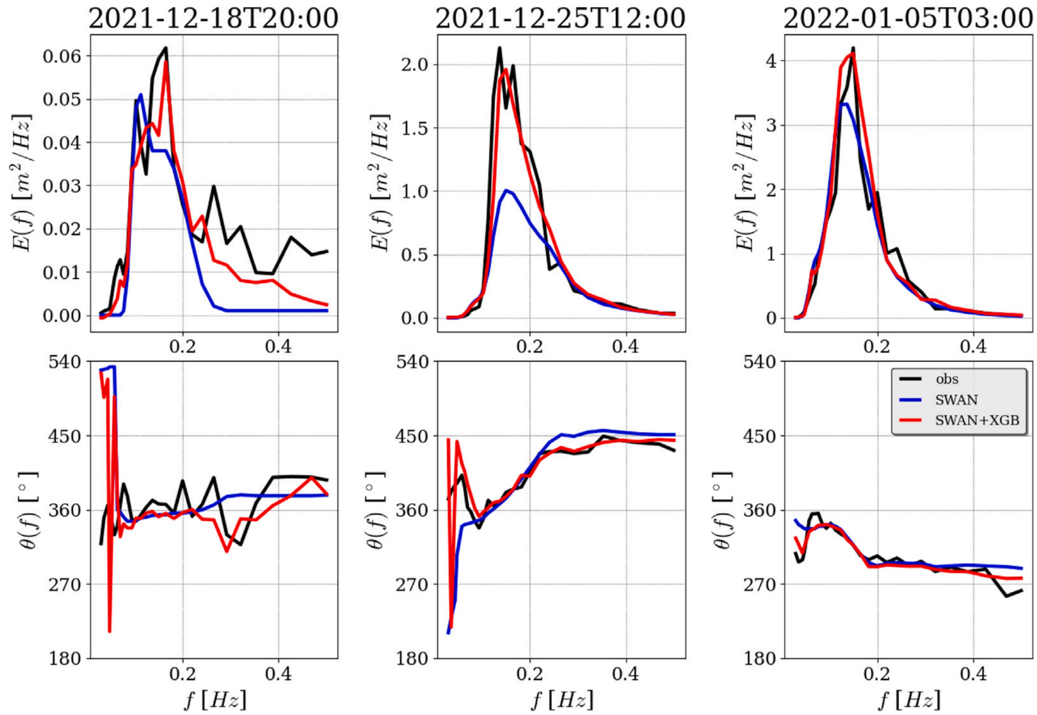
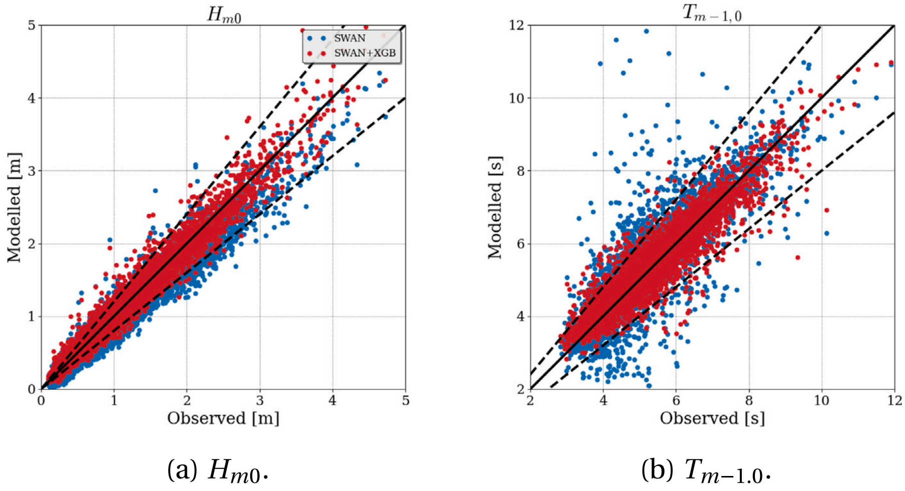


Fig. 4. Energy density spectra (upper panels) and directional spectra (lower panels) for location EPL for three moments in time, showing observed (black), SWAN (blue) and SWAN$_{corr}$ (red) predictions. Note that the directions are shown for a wider range (180 and 650°) to enhance visibility and avoid fluctuations between 360° and 0°.



(a) H_{m0}.

(b) $T_{m-1.0}$.

Fig. 5. Spectral wave parameters for the test data set as predicted by SWAN (blue) and SWAN$_{corr}$ including correction (red).

minimum number of data points and by applying an early stopping algorithm combined with a validation data set, as is done in this work.

In the model training process, about 18 months worth of data is used to train, validate and test the model. As long as both the operational wave model and the wave measurements are functional and their results are being stored, this combination is continuously generating new data that could benefit model training. Hence, it is expected that the ML models become progressively more accurate as the amount of data they are trained on grows larger. From a practical standpoint, very frequent retraining of the ML model is not expected to add a lot of value, since often occurring mild wave conditions should be well represented in the training data set already. Also, the training process has a relatively high computational demand, with training times (which depend on the computational resources used) varying from several hours to several days per model on the computational cluster used in this work. This high computational demand makes that retraining is preferably only attempted when a clear benefit to model performance is expected. However, severe wave events (e.g. storms) are much rarer than mild wave conditions, and therefore present a valuable addition to the training data. Thus a pragmatic approach might be to retrain the ML models after every storm season to ensure the largest impact from the retraining effort. Note that it is recommended that retraining is combined with a critical comparison of the performance of the newly trained models with that of the existing ones.

As mentioned above, there is an imbalance in the quantity of training data representing mild wave conditions (relatively calm seas) and severe wave conditions (storms), with the latter being far more rare than the former. In principle, it is then likely that a ML model will perform better for calm seas than for storms, simply because the larger number of data points represents a larger part of the error metric that is being minimized. If this is undesirable with regards to the foreseen application of the ML model, the use of differentiated weights in model training could be considered. By giving the smaller number of data points representing storms each a larger weight than those representing calm seas, the ML model will be inclined to converge towards a better performance for stormy conditions. The approach presented in this work is tailored towards being broadly applicable, hence no differentiated weighting was used in model training.

In practice an operational wave model might be improved or changed with some regularity, for instance with an updated bathymetry, newly calibrated model settings, or improved description of the physical processes in the wave model. For a ML model used in such an operational context, this raises the question whether a model trained on data generated before the change in the wave model is still valid to use after the change takes effect. There is no singular answer to this question, since it depends on the magnitude of the effect of the model change on the locations for which corrections are derived. Hence, it is often difficult to predict beforehand. One approach could be to retrain the ML model with only data generated after the change in the wave model. In this way, the ML model is certainly valid but at the cost of throwing away a large amount of potential training data, and this does not provide an alternative for the period between the wave model change and the time enough data is available to train the ML model on. Alternatively, the performance of the existing ML model can be compared between two test data sets, selected from data before and after the wave model change respectively. Comparable performance on both data sets implies that the existing ML model is still good enough to use, even after the change in the wave model.

As described in Section 2.4, the ML models are trained on the nowcast data. To also account for the uncertainty in the SWAN model forcing (wind and water level), there are two general directions that can be taken. One option is to construct a ML model to predict a correction for the wind velocity, wind direction, water level and current fields (similar to the approach improving the wind velocity field suggested by Yevnin and Toledo (2022)) and/or the offshore boundary conditions. These corrected values can then be used as forcing for the SWAN model. The subsequent SWAN predictions can then in turn be corrected with the ML model presented in this work. The other option is to construct one single model that uses multiple predictions with different lead times as input. For the latter direction, possibly Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) models could be used. These allow for the temporal dimension to be taken into account, so that the input can consist of a whole series of forecasts with different lead times. Any such exploratory research is outside the scope of the current work.

The approach presented in this work is based on discrete locations where both model predictions and wave measurements are available. This in turn means that the corrections are in principle only available on the same locations. Since the operational wave model is capable of predicting the whole wave field within its model domain, it begs the question whether it is also possible to derive a (spatially varying) correction for the entire model domain. One of the main difficulties is that the measurements are limited to a small number of discrete locations that do not cover the entire extent of the model domain. To gauge whether any attempt at a correction for the whole model domain performs well or not, a comparison with wave measurements over the same domain seems indispensable. Just adding more point measurements is likely not the solution, since the effects of local variations within the domain in for instance bathymetry will not be captured. A possible solution strategy could include trying to constrain the ML methods using prior physical knowledge, somewhat akin to the physics-informed neural networks proposed by Wang et al. (2022).

6. Conclusions

In this paper, an innovative hybrid modelling approach is proposed to improve operational wave forecasts. This approach is generically applicable to spectral wave models in an operational setting. In this work, the approach has been applied in the Dutch North Sea, where an operational SWAN wave model is combined with a XGBoost machine learning model that corrects both the energy density spectrum and directional spectrum predicted by SWAN. A total of 14 different locations is grouped into four regions along harbour approach channels, with a machine learning model being trained for each region. For these regions, around 18 months of data is available for training, validation and testing of the machine learning model.

When engineering the features of the input data, it becomes apparent that replacing strongly correlated features with uncorrelated ones (replacing wave period with wave steepness and wave direction with the difference between wind and wave direction) significantly reduces the computational effort posed by the model training. Subsequently, the hyperparameters of the machine learning model are tuned for both energy and direction for all regions. The differences between regions prove to be negligible, so one consistent set of hyperparameters can be used for all models.

The SWAN model including the XGBoost correction – SWAN$_{corr}$ – results in a significant average decrease in the root-mean-squared error compared to just the operational SWAN model, amounting to 21.7% for the wave energy density and 25.3% for the wave direction. The error reduction for the spectral wave parameters is even larger, with a 33.3% smaller error in spectral wave height and a 38.8% smaller error in spectral wave period. Hence, the SWAN$_{corr}$ model is shown to be a significant improvement of the operational wave model. As mentioned, its approach is generically applicable to spectral wave models, so it could also be applied to another model domain, another spectral wave model, or both. Additionally, with a slightly adapted approach the applicability does not need to be limited to wave models at all, and could be widely applied in operational contexts where one has measurement data and predictions on the same locations.

CRediT authorship contribution statement

Joost P. den Bieman: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Menno P. de Ridder:** Methodology, Software, Validation, Formal analysis, Data Curation, Writing – original draft, Writing – review & editing, Visualization. **Marisol Irías Mata:** Software, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Joana C.C. van Nieuwkoop:** Conceptualization, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

Declaration of competing interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in this paper is publicly available and the data sources have been referred to in the manuscript is available from public sources listed in the manuscript.

Acknowledgements

This research was funded by Rijkswaterstaat WVL through the Deltares programme ‘KPP Hydraulica schematisaties zout’ and by the Deltares research programme ‘Enabling Technologies’. Additionally, the authors wish to thank Jan Rolf Hendriks for his help in obtaining the wave data.

Appendix. Error metric

To show the performance of the different models, the root-mean-squared error (RMSE) is used as an error metric. The RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}} \tag{A.1}$$

where x_i is the modelled value, y_i the observed value and N the number of data points. To determine the error metric for circular datasets like the wave direction, 360° is added to the modelled values when the difference between the modelled and observed values is smaller than 180°. 360° is subtracted from the modelled values when the difference between the modelled and observed values is smaller than −180°. This additional step is required to compute the RMSE for circular datasets in a proper way.

References

Bengtsson, L., Andrae, U., Aspelien, T., Batrak, Y., Calvo, J., de Rooy, W., Gleeson, E., Hansen-Sass, B., Homleid, M., Hortal, M., Ivarsson, K.-I., Lenderink, G., Niemelä, S., Nielsen, K.P., Onvlee, J., Rontu, L., Samuelsson, P., Muñoz, D.S., Subias, A., Tijm, S., Toll, V., Yang, X., Køltzow, M.Ø., 2017. The HARMONIE–AROME model configuration in the ALADIN–HIRLAM NWP system. Mon. Weather Rev. 145 (5), 1919–1935. http://dx.doi.org/10.1175/mwr-d-16-0417.1.

Booij, N., Holthuijsen, L.H., Ris, R.C., 1996. The "SWAN" wave model for shallow water. In: Coastal Engineering 1996. pp. 668–676. http://dx.doi.org/10.1061/9780784402429.053.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Callens, A., Morichon, D., Abadie, S., Delpey, M., Liquet, B., 2020. Using random forest and gradient boosting trees to improve wave forecast at a specific location. Appl. Ocean Res. 104, 102339. http://dx.doi.org/10.1016/j.apor.2020.102339.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. CoRR abs/1603.02754, arXiv:1603.02754.

de Ridder, M.P., den Bieman, J.P., Gautier, C., van Nieuwkoop, J.C.C., 2021. Machine learning improves the modelled wave spectrum in the north sea. In: Coastal Dynamics 2021 Conference. URL: http://resolver.tudelft.nl/uuid:b57ca601-eb1f-43ce-a5fa-5545157155b8.

den Bieman, J.P., van Gent, M.R.A., van den Boogaard, H.F.P., 2021. Wave overtopping predictions using an advanced machine learning technique. Coast. Eng. 166, 103830. http://dx.doi.org/10.1016/j.coastaleng.2020.103830.

ELI5, 2020. ELI5 Python package. URL: https://github.com/TeamHG-Memex/eli5.

Emmanouil, S., Aguilar, S.G., Nane, G.F., Schouten, J.-J., 2020. Statistical models for improving significant wave height predictions in offshore operations. Ocean Eng. 206, 107249. http://dx.doi.org/10.1016/j.oceaneng.2020.107249.

Fisher, A., Rudin, C., Dominici, F., 2018. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. arXiv:1801.01489.

Gautier, C., van Nieuwkoop, J.C.C., de Ridder, M.P., 2018. SWAN-Kuststrook. Technical Report, Deltares, 11202221-005-ZKS-0005.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Lopez, M., Lopez, I., Iglesias, G., 2015. Hindcasting long waves in a port: An ANN approach. Coast. Eng. J. 57 (04), 1550019. http://dx.doi.org/10.1142/S0578563415500199.

Mittendorf, M., Nielsen, U.D., Bingham, H.B., 2022. Data-driven prediction of added-wave resistance on ships in oblique waves - A comparison between tree-based ensemble methods and artificial neural networks. Appl. Ocean Res. 118, 102964. http://dx.doi.org/10.1016/j.apor.2021.102964.

Rijkswaterstaat, 2022. Multifunctional access tool for operational oceandata services (MATROOS). URL: https://matroos.rws.nl/.

The Wamdi Group, 1988. The WAM model-A third generation ocean wave prediction model. J. Phys. Oceanogr. 18 (12), 1775–1810.

Tsai, C.-P., Lin, C., Shen, J.-N., 2002. Neural network for wave forecasting among multi-stations. Ocean Eng. 29 (13), 1683–1695. http://dx.doi.org/10.1016/S0029-8018(01)00112-3.

van Rossum, G., 1995. Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI).

Wang, N., Chen, Q., Chen, Z., 2022. Reconstruction of nearshore wave fields based on physics-informed neural networks. Coast. Eng. 176, 104167. http://dx.doi.org/10.1016/j.coastaleng.2022.104167.

Wei, Z., Davison, A., 2022. A convolutional neural network based model to predict nearshore waves and hydrodynamics. Coast. Eng. 171, 104044. http://dx.doi.org/10.1016/j.coastaleng.2021.104044.

Yevnin, Y., Chorev, S., Dukan, I., Toledo, Y., 2023. Short-term wave forecasts using gated recurrent unit model. Ocean Eng. 268, 113389. http://dx.doi.org/10.1016/j.oceaneng.2022.113389.

Yevnin, Y., Toledo, Y., 2022. A deep learning model for improved wind and consequent wave forecasts. J. Phys. Oceanogr. http://dx.doi.org/10.1175/JPO-D-21-0280.1.