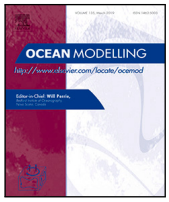
Ocean Modelling 164 (2021) 101832

Contents lists available at ScienceDirect

Ocean Modelling

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



Predicting Lake Erie wave heights and periods using XGBoost and LSTM

Haoguo Hu [a], André J. van der Westhuysen [b,*], Philip Chu [c], Ayumi Fujisaki-Manome [a]

[a] Cooperative Institute for Great Lakes Research, University of Michigan, Ann Arbor, MI, USA
[b] IMSG at National Oceanic and Atmospheric Administration, National Centers for Environmental Prediction, Environmental Modeling Center, College Park, MD, USA
[c] National Oceanic and Atmospheric Administration, Great Lakes Environmental Research Laboratory, Ann Arbor, MI, USA

ABSTRACT

Waves in large lakes put coastal communities and vessels under threat, and accurate wave predictions are needed for early warnings. While physics-based numerical wave models such as WAVEWATCH III (WW3) are useful to provide spatial information to supplement in situ observations, they require intensive computational resources. An attractive alternative is machine learning (ML) methods, which can potentially improve the performance of numerical wave models, while only requiring a small fraction of the computational cost. In this study, we applied novel ML methods based on XGBoost and a Long Short-Term Memory (LSTM) recurrent neural network for predicting wave height and period under the near-idealized wave growth conditions of Lake Erie. Data sets of significant wave height (H), peak wave period (T_p) and surface wind from two offshore buoys from 1994 to 2017 were processed for model training and testing. We trained and validated the ML models with the data sets from 1994 to 2015, and then used the trained models to predict significant wave height and peak period for 2016 and 2017. The XGBoost model yielded the best overall performance, with Mean Absolute Percentage Error (MAPE) values of 15.6%–22.9% in H and 8.3%–13.4% in T_p. The LSTM model yielded MAPE values of 23.4%–30.8% in H and 9.1%–13.6% in T_p. An unstructured grid WW3 applied to Lake Erie yielded MAPE values of 15.3%–21.0% in H and 12.5%–19.3% in T_p. However, WW3 underestimated H and T_p during strong wind events, with relative biases of -11.76% to -14.15% in H and -15.59% to -19.68% in T_p. XGBoost and LSTM improve on these predictions with relative biases of -2.56% to -10.61% in H and -8.08% to -10.13% in T_p. An ensemble mean of these three models yielded lower scatter scores than the members, with MAPE values of 13.3%–17.3% in H and 8.0%–13.0% in T_p, although it did not improve the bias. The ML models ran significantly faster than WW3: For this 2-year run on the same computing environment, WW3 needed 24 h with 60 CPUs, whereas the trained LSTM needed 0.24 s on 1 CPU, and the trained XGBoost needed only 0.03 s on 1 CPU.

1. Introduction

Accurate predictions of wave conditions are important for the offshore industry, shipping, and for mitigating coastal hazards such as dune erosion. Various wave models have been developed and implemented in ocean and coastal regions. Following Mahjoobi and Etemad-Shahidi (2008) and Pirhooshyaran et al. (2020), these can be divided into three broad categories, namely (i) physics-based models, which feature parameterizations of the wave action balance equation, (ii) statistical and machine learning models (hereafter ML models, also known as soft computing methods) which instead use the data structure for prediction, without explicit description of physics, and (iii) hybrid approaches, which combine the first two categories.

Physics-based models have been developed during the course of three generations, starting with simple parameterizations of the relationships between wind, wave height and wave period, to the current third generation in which four-wave nonlinear interaction and various other source terms are explicitly modeled in frequency-directional space, e.g. WAM (WAMDI Group, 1988), SWAN (Booij et al., 1999) and WAVEWATCH III (hereafter WW3, Tolman et al., 2002). For Lake Erie, one of the five Great Lakes, first- and second-generation physics-based wave models used simple parameterization to account for the evolution of wave height and period due to wind forcing (Schwab et al., 1984). Third-generation wave models such as SWAN and MIKE21 SW (Sørensen et al., 2004), which explicitly represent three- and four-wave interactions, have subsequently been applied to Lake Erie (Moeini and Etemad-Shahidi, 2007). A coupled wave-hydrodynamic model based on the unstructured grid Finite Volume Community Ocean Model FVCOM (Chen et al., 2013) and a third-generation wave model based on SWAN (Qi et al., 2009) was applied to Lake Erie to investigate wave climatology and inter-basin wave interactions (Niu and Xia, 2016). Alves et al. (2014) implemented the third-generation WW3 model as the dynamical core of the National Oceanic and Atmospheric Administration's (NOAA) Great Lakes operational wave forecasting system.

While physics-based wave models generally provide satisfactory results in comparison with observations, the physics they embody are known to underperform under highly nonlinear, extreme events such as storm peaks (e.g. Alves et al., 2014). Furthermore, a practical problem, especially in an operational environment, is their high computational expense.

Statistical and ML models (soft computing methods) were first applied to significant wave height prediction by Deo and Sridhar Naidu (1999), Deo et al. (2001) and Agrawal and Deo (2002). These researchers applied small, fully-connected, feed-forward artificial neural

* Corresponding author.
E-mail address: andre.vanderwesthuysen@noaa.gov (A.J. van der Westhuysen).

https://doi.org/10.1016/j.ocemod.2021.101832
Received 22 November 2020; Received in revised form 18 May 2021; Accepted 2 June 2021
Available online 8 June 2021

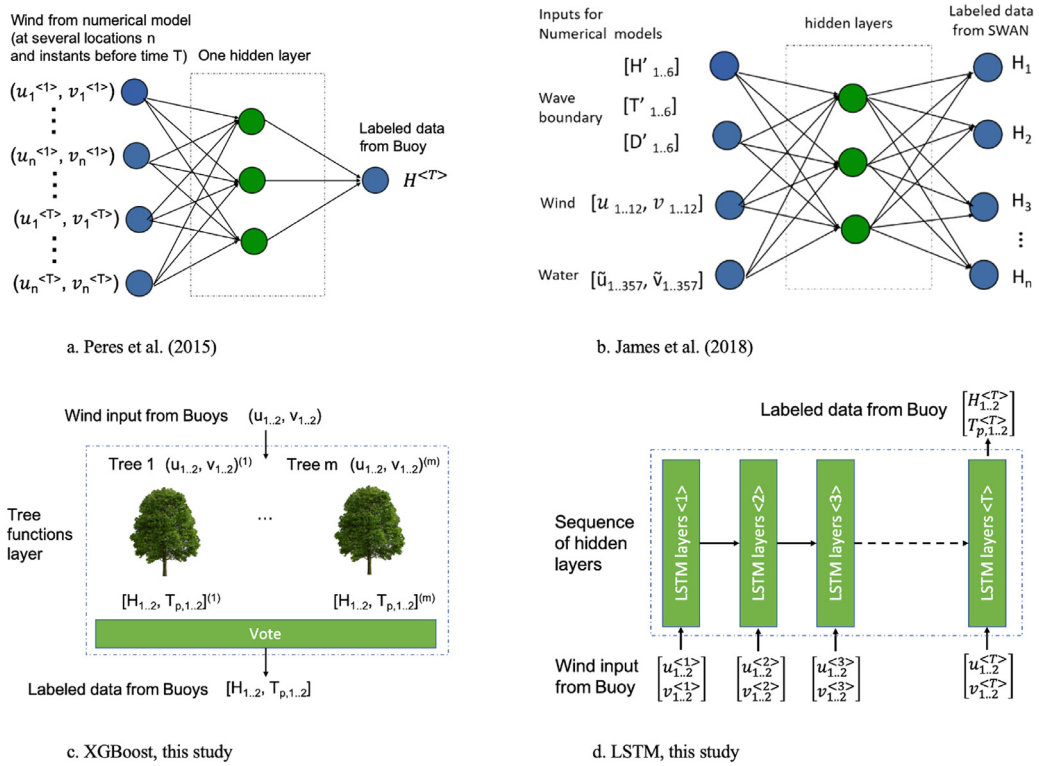


Fig. 1. Schematics of machine learning models considered in this study. Input or output locations denoted by subscripts. Time levels denoted by superscripts in brackets. Input samples used in ensemble members denoted by superscripts in parentheses.

networks (ANNs), as well as autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) stochastic time series models to predict waves at point locations using as input previous wave conditions (univariate modeling) or wind observations (multivariate modeling). Mandal and Prabaharan (2006) introduced the use of recurrent neural networks (RNNs) to wave forecasting. They applied an RNN to predict significant wave heights based on a time series history of observed significant wave heights. More recently, Srinivasan et al. (2017) compared feed-forward networks and RNNs in the context of multivariate wave modeling, using observed wind as input, and found the RNNs to have better skill. All these studies applied neural networks with only one hidden layer, and limited data sets of only 1–2 years.

Mahjoobi and Etemad-Shahidi (2008) and Etemad-Shahidi and Mahjoobi (2009) introduced the use of single classification and regression trees (using C5.0, CART and M5' training algorithms) as an alternative to neural network-based approaches. The former study considered 5 years of wind and wave data on Lake Michigan from the National Data Buoy Center (NDBC) station 45007. The authors found better performance for regression trees than classification trees, and that ANNs performed slightly better than either tree model. The authors note that, as with ANNs, tree models do not require knowledge of the underlying physical process to make predictions. However, they provide more insight into the problem than ANNs, since they represent understandable rules. Mahjoobi et al. (2008) and Malekmohamadi et al. (2011) compared a range of other ML approaches, including Support Vector Machines, Bayesian Networks, Fuzzy Inference Systems, and Adaptive Neuro-Fuzzy Inference Systems to ANNs in Lake Ontario and Lake Superior. They found comparable accuracy between these alternative methods and ANNs, except for Bayesian networks, which performed poorly. However, it should be noted that the wind and wave data used in their study covered less than two years, and the ANN used had only one hidden layer with three neurons.

Recent years have seen breakthroughs in the development of ML training algorithms, hardware, and a massive increase in data. Peres et al. (2015) used observed winds (u, v) at or near three coastal wave buoy stations as input, and observed significant wave heights (H) as output, to train a single hidden-layer, fully-connected, feed-forward neural network for each station (Fig. 1a). The data set covered a 20-year period from 1989 to 2008. The model input included observed wind from locations upwind of the output station as well as at time lags, to account for the fetch and duration in wind-wave generation. James et al. (2018) trained a fully-connected, feed-forward neural network to produce a field of coastal wave heights H_1–H_n, using as input three offshore wave characteristics (height H, period T, and direction D), a 12 × 2 field of wind vectors (u, v), a 357 × 2 field of ocean currents ($\tilde{u}$, $\tilde{v}$), and wave height fields from SWAN simulations as target variables (Fig. 1b). Recently, Feng et al. (2020) used fully-connected, a feed-forward neural network to model waves in Lake Michigan. A number of more complex neural network applications have furthermore been proposed: Mahmoodi et al. (2017) compared feed-forward and cascade-forward networks, but found better performance in the former. Kumar et al. (2018) investigated the use of ensembles of Extreme Learning Machines to reduce the variation caused by random initialization. Prahlada and Deka (2015) and Dixit and Londhe (2016) combined ANNs with wavelet analysis to better model extreme wave events. Pirhooshyaran and Snyder (2020) introduced the use of sequence-to-sequence networks for the prediction wave height and output power, feature selection, and the reconstruction of missing wave observations using data from nearby stations. Pirhooshyaran et al. (2020) expanded this work by including an ensemble of sequence-to-sequence networks, introducing a new resampling technique and improving feature selection. They demonstrate the superiority of this approach as the forecast lead time increases.

Hybrid approaches combine the strengths of physics-based models and ML. This includes the use of ML models to perform data assimilation on the output of physics-based models (e.g. Makarynskyy, 2006; Londhe et al., 2016), and the creation of nonlinear ensemble means (e.g. Campos et al., 2020).

In the present study, we investigate two different ML approaches for learning the nonlinear relationships between observed wind data,

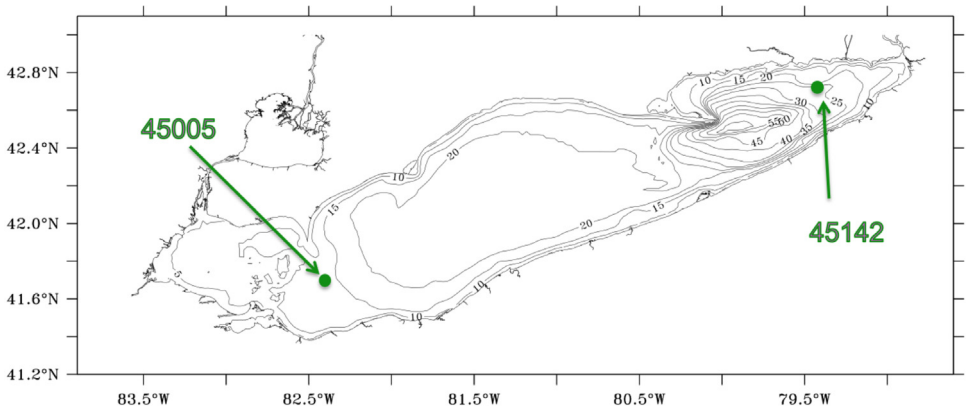


Fig. 2. Lake Erie bathymetry shown by contour lines (meters). Green dots are the NDBC buoys 45005 and 45142.

significant wave height and peak wave period, with the aim to correct the underprediction of the latter two quantities by physics-based spectral wave models, in particular at storm peaks. The first is an ensemble tree-based model, XGBoost (Chen and Guestrin, 2016), which is based on the Random Forest model (Breiman, 2001). XGBoost is an optimized, distributed gradient boosting library designed to be highly efficient, flexible, and portable. It provides a parallel tree boosting that solves many data science problems, such as regression (e.g. Qian et al., 2020) and classification (e.g. Merembayeva et al., 2019), fast and accurately. XGBoost has built-in L1 and L2 regularization which prevents the model from overfitting. It is furthermore robust, having a capability to handle missing values. The second model is from the RNN family, used here to capture the dependency of significant wave height and peak wave period on the time series structure of recently-observed winds. Specifically, we applied the Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber, 1997), implemented using the Keras high-level framework with a TensorFlow backend (Chollet, 2017). To compare with physics-based models, we include simulations with the unstructured mode of WW3 (Roland and Ardhuin, 2014). The field case considered here is Lake Erie, which is an enclosed basin, with a wind climate directed along its major axis. Hourly metocean data are available at NDBC stations 45005 and 45142 at opposite ends of the fetch, for the period 1994–2017. It therefore represents near-idealized conditions under which to compare the modeling of wind-wave growth by ML and physics-based models. The XGBoost (Fig. 1c) and LSTM (Fig. 1d) models were trained using observed wind (u, v), significant wave height H and peak wave period T_p at these two data buoys (Fig. 2), and subsequently evaluated against holdout data and the physics-based model WW3.

This paper is structured as follows: Section 2 describes the field case and data set used to train and evaluate the two ML models. Sections 3 and 4 provide details of respectively the XGBoost and LSTM models. Section 5 describes the physics-based WW3 model and the configuration applied here. In Section 6, the performance of the two ML models and WW3, as well as their ensemble mean, are intercompared, and in Section 7 model sensitivity is investigated. Section 8 closes with conclusions drawn from these results.

2. Field case data set

The field case considered here for the training and evaluation of our ML models is Lake Erie, one of the Great Lakes that is shared between the United States and Canada (Fig. 2). Lake Erie is narrow, with a length of 338 km and a width of 92 km, and has a WSW-ENE orientation. It is relatively shallow, with a maximum depth of 64 m, but with mean depths of only 7.3 m in the western basin, 18.3 m in the central basin, and 24.4 m in the eastern basin. Lake Erie has several offshore buoys as part of its observational network, of which two have collected continuous hourly data from 1994 to 2017 during the ice-free spring, summer and fall seasons. Located on the western end of the lake, near Toledo, OH is station 45005 (9.8 m depth), maintained by NOAA's National Data Buoy Center (NDBC). Located on the eastern end of the lake, near Buffalo, NY is station 45142 (27 m depth), maintained by the Environment and Climate Change Canada. The historical and real-time measurements from these two offshore buoys are available from NDBC (2020). This data was cleaned by (i) dropping records where the wind speed, significant wave height and peak wave period were not available simultaneously (as required for training and evaluation), (ii) dropping records where the significant wave height is very low (<0.05 m) indicating low-energy noise, and (iii) dropping records where the peak wave period had a quality flag of > 1 (indicating doubtful or erroneous values) or an unphysical value for this small lake (>14 s). In addition, based on visual inspection, some years were excluded due to the doubtful quality of the peak period observations at station 45142, namely 1996, 2003–2006, and 2010.

Figs. 3 to 5 show the distributions of the cleaned monthly significant wave height, peak wave period, and wind speed during the 24-year observational period considered. Fig. 6 shows the observed wind climate in terms of wind roses at NDBC buoys 45005 and 45142. From Fig. 3 we can see that the distribution of significant wave heights is strongly skewed to the right (higher values), with the third quartile below 1.5 m, but with outliers (representing storm peaks) of up to 5.2 m. There is clear seasonality present, with the largest significant wave heights recorded in spring (April–May) and fall (October–November). Similarly, Fig. 4 shows peak wave periods to be skewed to the right and display seasonality, with the highest values recorded in the spring and fall. Fig. 5 shows that this is correlated to similar seasonal wind speed magnitude patterns, which is to be expected considering that all wind wave conditions in this shallow lake are locally generated. Figs. 3 and 4 furthermore show that extreme significant wave heights and peak periods tend to be higher at station 45142 than station 45005. Considering that the dominant wind direction is from the WSW-SSW (Fig. 6), these larger wave heights are due to the longer fetch available to the downwind station 45142. Note that the extreme depth limitation of 7.3 m at station 45005 also imposes a depth limitation to the maximum possible wave height, which is relaxed at the deeper buoy 45142.

The input variables that were selected to construct our ML models are firstly the observed wind speed and direction at stations 45005 and 45142. These are transformed into the wind vector components u and v at each station. Second, following Peres et al. (2015), we add lags of these wind vectors to capture storm duration. Considering the 338 km length of the lake, and a maximum observed wind speed of 20.6 m/s during the study period, fully developed waves would be generated after a duration of about 24 h (following Bretschneider, 1958). Therefore, lags at 3-hourly intervals of up to 24 h were included. The exact number of lags included was determined as part of hyperparameter

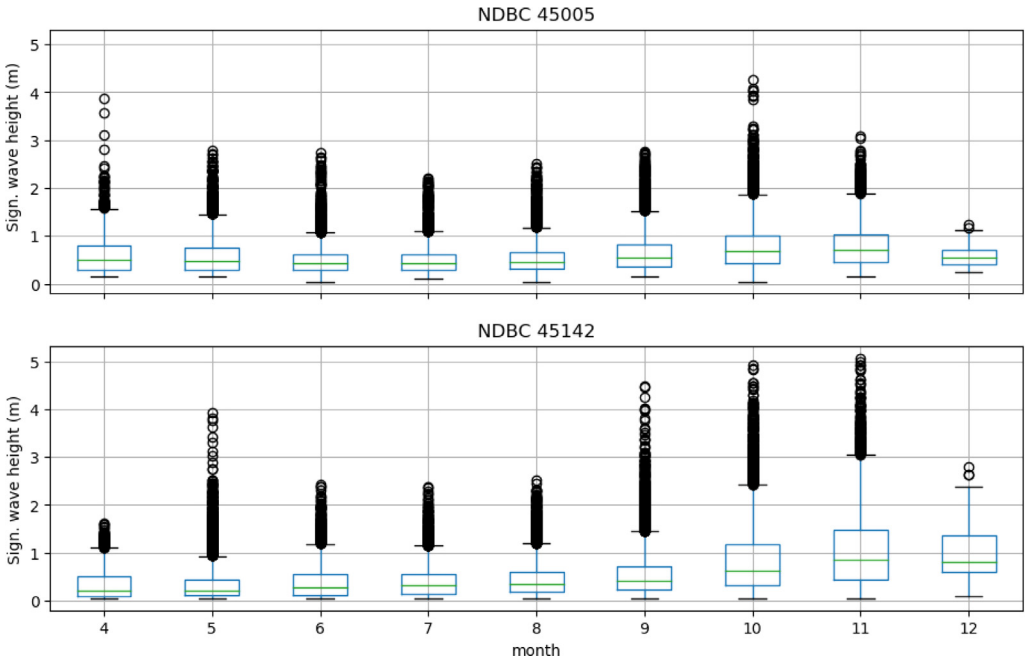


Fig. 3. Monthly distributions of significant wave height during the ice-free months (April–December). Top panel: NDBC 45005; Bottom panel: NDBC 45142.

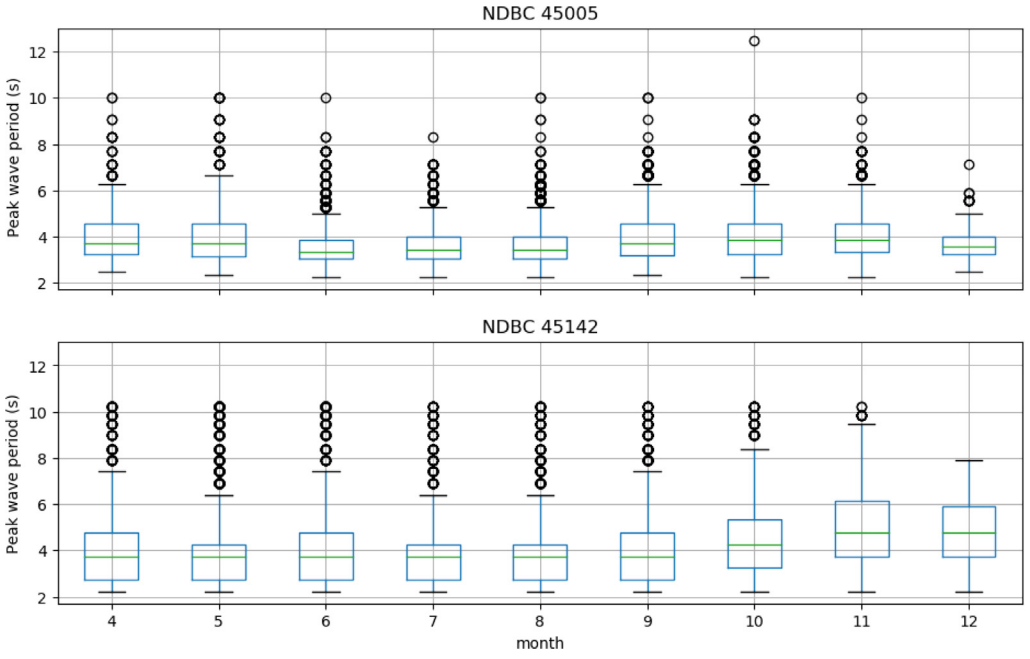


Fig. 4. Monthly distributions of peak wave period during the ice-free months (April–December). Top panel: NDBC 45005; Bottom panel: NDBC 45142.

tuning (see below). This yields a maximum of 18 input variables per observation station, or 18 × 2 in total. The observed significant wave height and peak wave period at stations 45005 and 45142 are used as the output (target) variables. Note that in general wave conditions are also affected by lake ice coverage and thickness, and hence lake surface temperature, which could serve as additional input variables. However, here we only consider wave growth during ice-free months (April to early December), so that these variables are not included.

In order to train and evaluate our ML models on this data, we split the data set into training, validation and test partitions. Considering that these are time series data, in which serial correlation exists between successive observations, the usual approach of random assignment to the three partitions is not followed. Instead, continuous periods of the time series are assigned to each partition (e.g. Peres et al., 2015). The bulk of the data set, namely the years 1994–2012, is assigned to the training partition, in order to achieve reliable estimates of the trainable model parameters. Next, the years 2014–2015 are assigned to the validation partition, which is used in the selection of model hyperparameters such as tree depth and to check for overfitting on the training data. Finally, the years 2016–2017 are assigned to the test partition. This latter portion of the data set will be used to test how well the ML models generalize to unseen conditions at the two locations for which they were trained. The test partition will also be used to assess the performance of the physics-based model WW3 at these two stations. The resulting split between the three partitions is 28 154 records for training, 5 193 for validation, and 4 493 for testing,

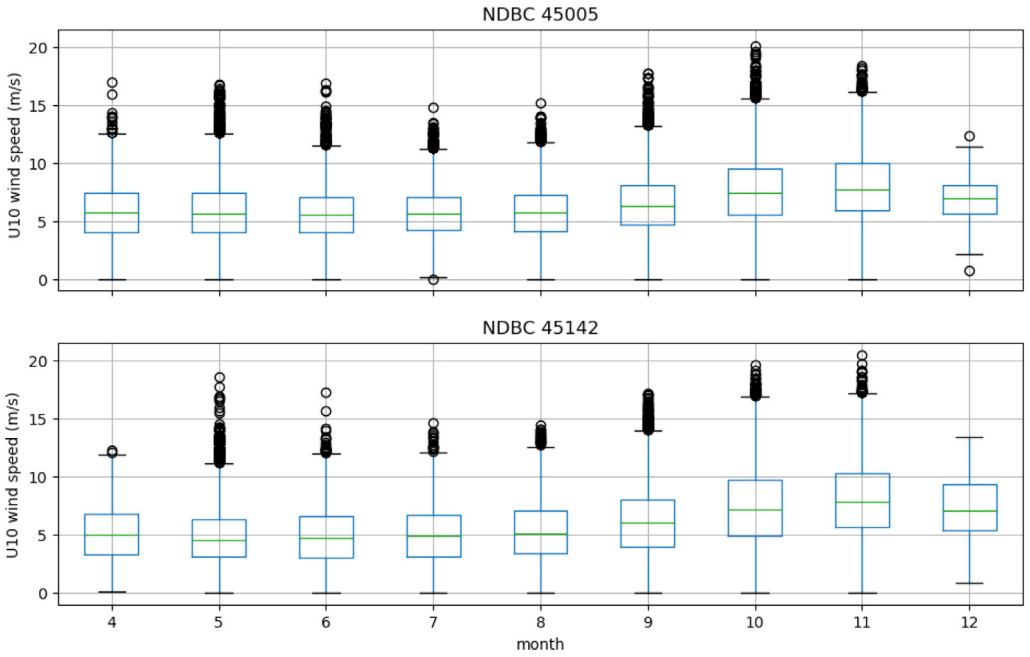


Fig. 5. Monthly distributions of U_{10} wind speed during the ice-free months (April–December). Top panel: NDBC 45005; Bottom panel: NDBC 45142.

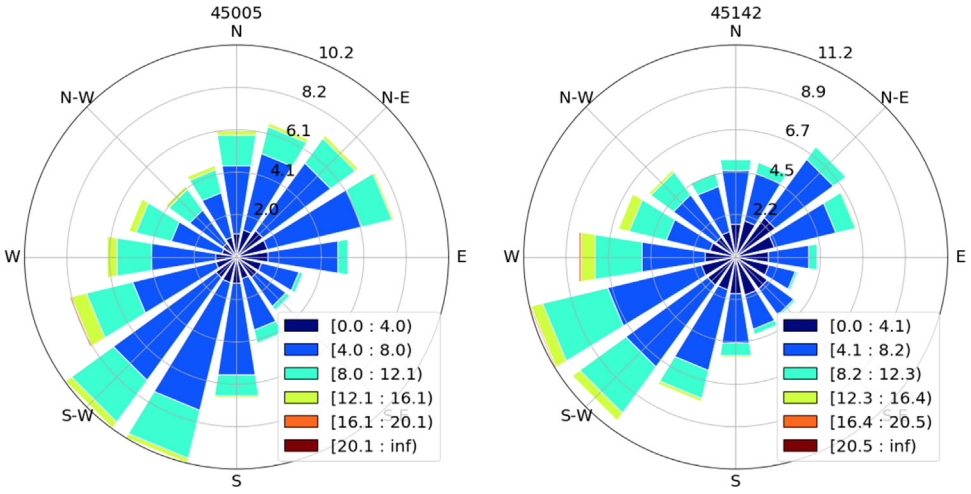


Fig. 6. Wind roses for NDBC 45005 (left) and NDBC 45142 (right). Color classes indicate wind speed (m/s). Radial distance indicates percentage occurrence.

or about 74%:14%:12%. Fig. 7 compares the distribution of observed significant wave height and peak wave period in the train, validation and test partitions at stations 45005 and 45142. We can see that the median and interquartile distributions are similar between the training, validation and test partitions. However, we note that in terms of peak events (outliers), the test partition is less energetic than the train and validation partitions.

3. XGBoost model

3.1. Model description

XGBoost (Chen and Guestrin, 2016) is mostly used for supervised learning problems, where one uses training data to predict a target variable. XGBoost implements ML algorithms under the Gradient Boosting framework. It uses the same tree ensemble model as the Random Forest model (Breiman, 2001), consisting of a set of classification or regression trees. Random Forest and boosted trees are basically the same models; the only difference is how one trains them.

Mathematically, an ensemble tree model can be written in the form (e.g. Chen and Guestrin, 2016):

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in F \tag{1}$$

where K is the number of trees, and $f(\mathbf{x})$ is a tree function in the function space $F = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$. The data set $D = \{(\mathbf{x}_i, y_i)\}$ has n examples and m features ($\mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}$). The structure of each tree is given by q, which maps an example $\mathbf{x}_i$ to the corresponding leaf index. Each tree function f_k corresponds to a structure q and leaf weights w, for a total of T leaves.

The objective/loss function with parameters θ to be optimized is given by

$$L(\theta) = \sum_i l\left(y_i, \hat{y}_i\right) + \sum_k \Omega(f_k) \tag{2}$$

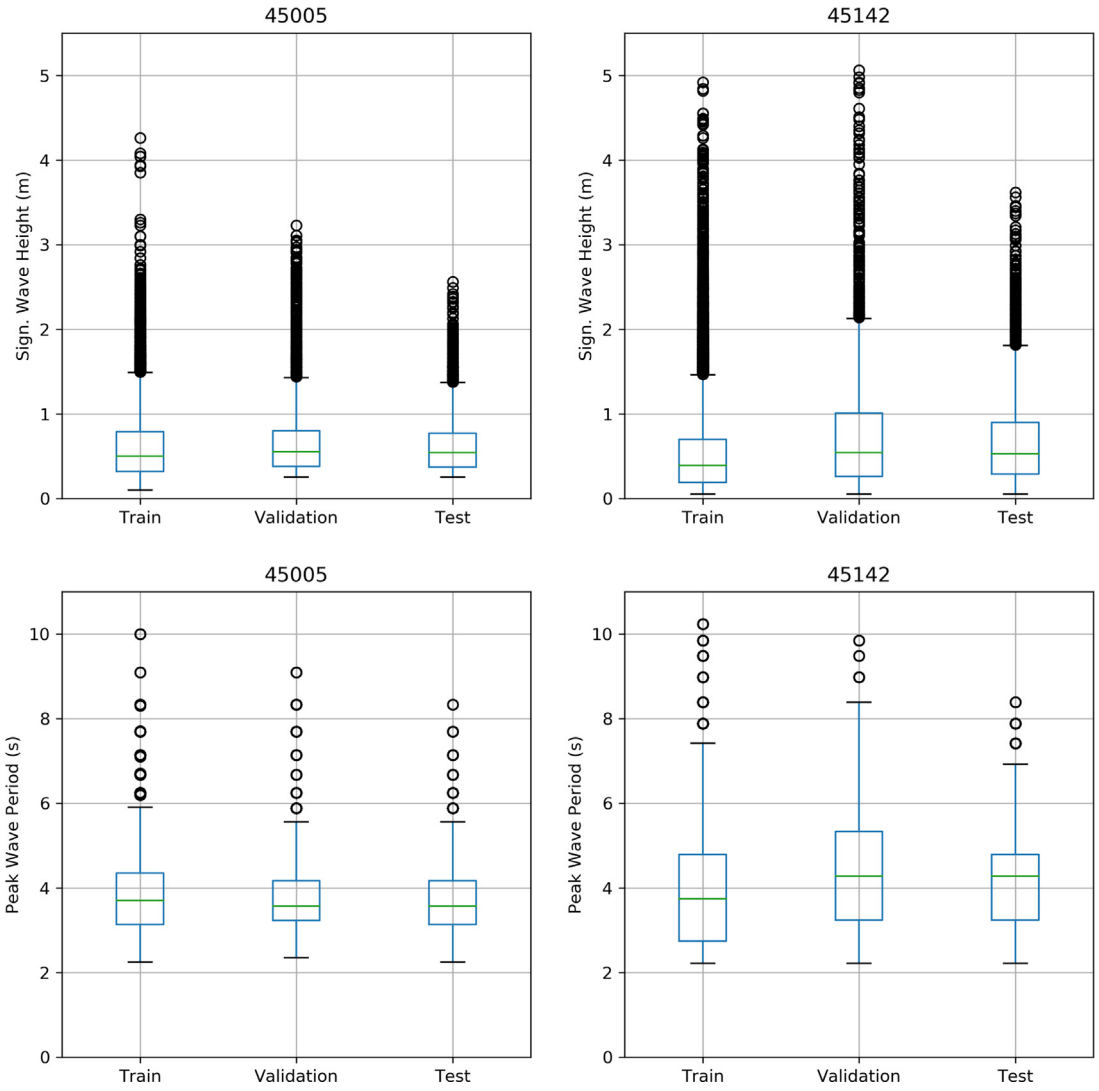


Fig. 7. Distribution of significant wave height and peak wave period in the training, validation, and test partitions at NDBC stations 45005 (left) and 45142 (right).

where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. $\Omega(f)$ is the regularization term, which penalizes the complexity of the model.

Gradient tree boosting is achieved by training the model in an additive manner—tree functions f_t are greedily added to successive iterations to minimize the loss function (2). Let $\hat{y}_i^{t-1}$ and $\hat{y}_i^t$ be the predictions at iterations $(t-1)$ and t, respectively, of the ith instance. Setting $\hat{y}_i^t = \hat{y}_i^{t-1} + f_t(\mathbf{x}_i)$, the loss function at the tth iteration will be:

$$L^t = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{t-1} + f_t(\mathbf{x}_i)\right) + \Omega(f_t) \tag{3}$$

Recalling the Taylor expansion $f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$, we obtain

$$l\left(y_i, \hat{y}_i^{t-1} + f_t(\mathbf{x}_i)\right) = l\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \tag{4}$$

Then we have

$$L^t \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t) + \text{constant} \tag{5}$$

where $g_i = \partial_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right)$ and $h_i = \partial^2_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right)$.

Define the penalty on the complexity of the model with the regularization term $\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$, with T the number of leaves. Then, with constant terms removed, the simplified objective (loss function) at iteration t is:

$$L^t \simeq \sum_{i=1}^{n} \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{6}$$

Define the instance set of leaf j as $I_j = \left\{ i | q(\mathbf{x}_i) = j \right\}$, then

$$L^t \simeq \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{7}$$

Define $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$ then

$$L^t \simeq \sum_{j=1}^{T} \left[G_j w_j + \frac{1}{2} \left(H_j + \lambda \right) w_j^2 \right] + \gamma T \tag{8}$$

And then

$$\underset{\mathbf{x}}{\text{argmin}}\,(L) = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{9}$$

where the optimal weight w_j^* of leaf j is given by $w_j^* = -\frac{G_j^2}{H_j + \lambda}$.

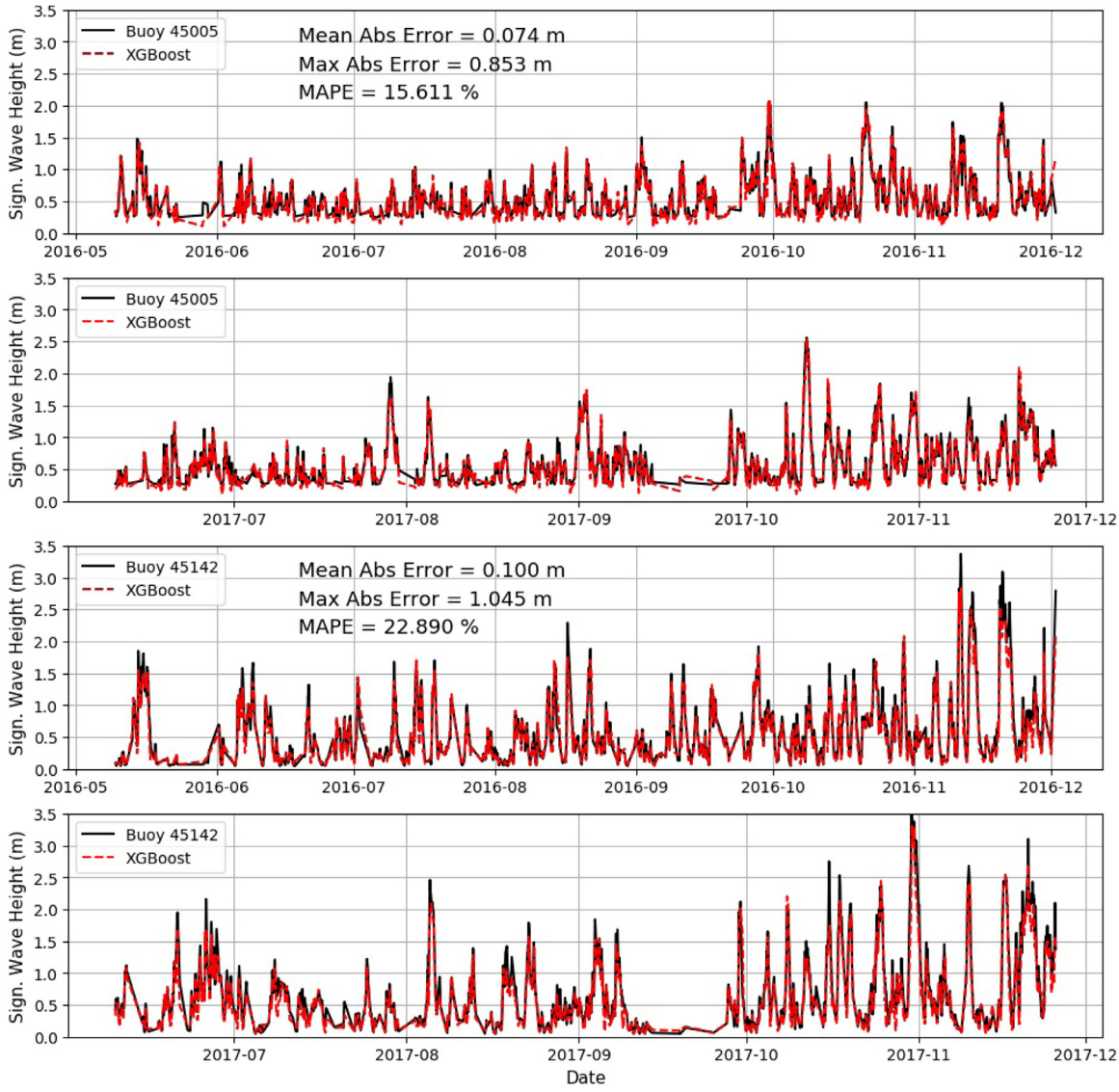


Fig. 8. Time series of the XGBoost significant wave height predictions for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

In summary, the boosted tree algorithm is the following:

1. Add a new tree in each iteration.
2. At the beginning of each iteration, calculate: $g_i = \partial_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right)$ and $h_i = \partial^2_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right)$.
3. Use the statistics to greedily grow a tree $f_t(\mathbf{x})$ with $L = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j+\lambda} + \gamma T$.
4. Add $f(\mathbf{x})$ to the model $\hat{y}_i^t = \hat{y}_i^{t-1} + \varepsilon f_t(\mathbf{x}_i)$, where ε is called shrinkage.

3.2. Configuration and hyperparameter tuning

The XGBoost model used hourly wind, significant wave height and peak wave period from observations at the buoy stations in western (45005) and eastern (45142) Lake Erie (Fig. 2). The observed wind speed and direction at the two stations were used as input X_i, and significant wave height and peak wave period were used as model output y_i. For this regression problem, we set the learning objective as regression with squared loss.

Three hyperparameters were optimized in the XGBoost model, namely: TD = Maximum Tree Depth, W = Minimum Child Weight, and E = learning rate (eta). Considering the limited number of hyperparameters, a direct search strategy was followed for the optimization: The search space was defined as $TD = [5, 6, 7, 8, 9, 10, 11, 12]$, $W = [3, 4, 5, 6, 7, 8]$, and $E = [0.005, 0.01, 0.05, 0.1, 0.2]$. One parameter was tuned at a time (with other parameters fixed) until all parameters were tuned. The process was repeated three times. Mean Absolute Percentage Error (MAPE) was set as the metric for training and the Mean Absolute Error (MAE) as the evaluation metric for cross validation. Table 1 shows that model accuracy improved with increasing maximum tree depth TD up to 10, but was quite insensitive to the minimum child weight W. The combination of TD=10 and $W = 5$ yielded a minimum MAE = 0.07466 m. Table 2 shows that the optimal learning rate $E = 0.05$ yielded a minimum MAE = 0.0747 m. This model training ran on a CPU (Intel® Xeon® CPU E2176M @ 2.70 GHz) for about 3 min per hyperparameter combination, and it converged at around 4 000 epochs.

3.3. XGBoost results

The trained XGBoost model was used to predict wind-wave conditions for the unseen test period of 2016–2017. It needed only 0.03 s for this prediction, using 1 CPU. Figs. 8 and 9 show the time series and error statistics. The prediction for station 45005 yielded a MAE

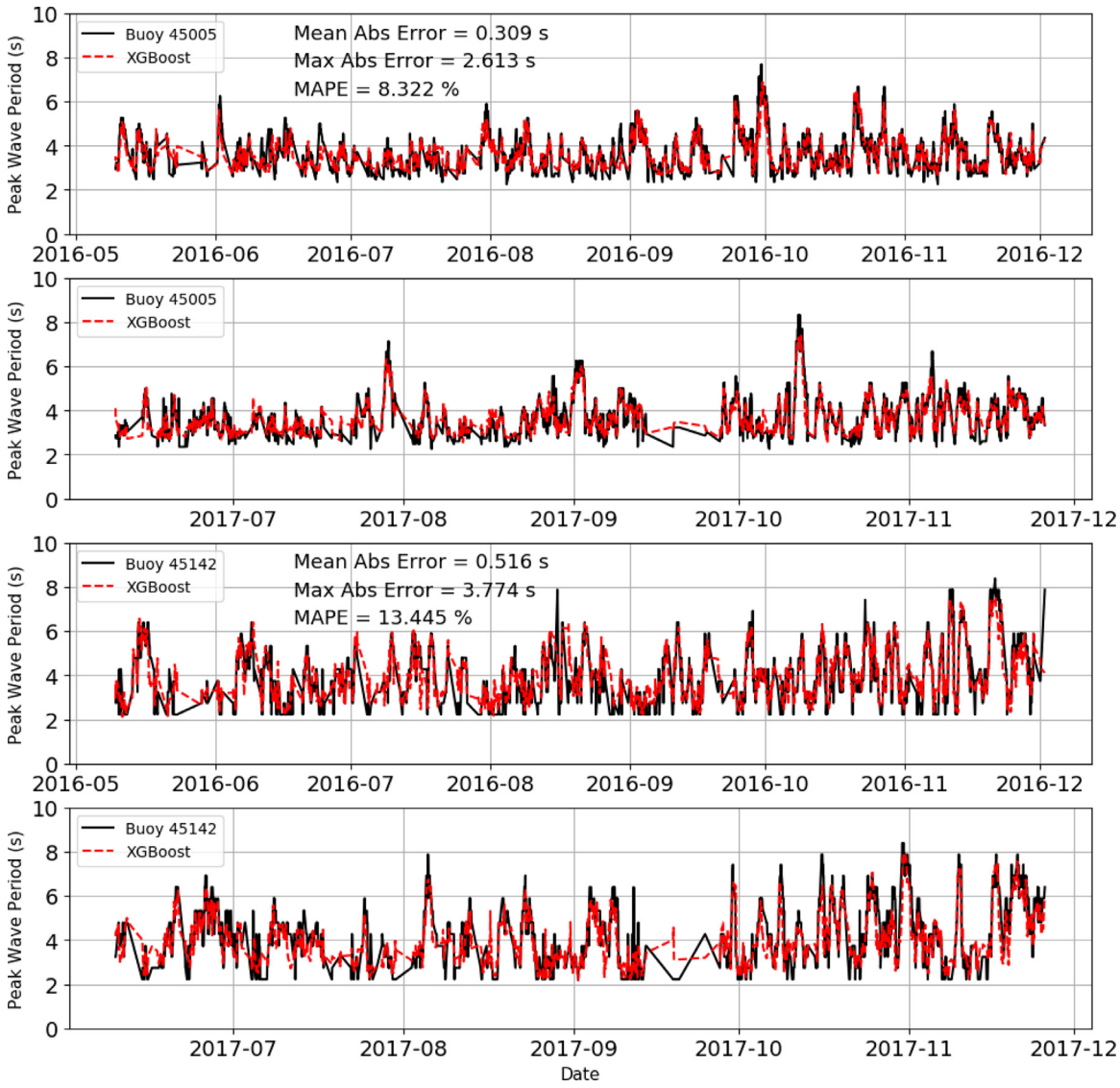


Fig. 9. Time series of the XGBoost peak wave period predictions for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

Table 1
Hyperparameter tuning results on the validation partition (MAE, m) for XGBoost along the dimensions of TD = Max Tree Depth and W = Minimum Child Weight, for a fixed optimal learning rate E = 0.05. Optimal settings shown in bold.

W	TD							
	5	6	7	8	9	10	11	12
3	0.07672	0.07605	0.07552	0.07508	0.07492	0.07475	0.07482	0.07495
4	0.07673	0.07606	0.07551	0.07508	0.07493	0.07471	0.07475	0.07475
5	0.07675	0.07605	0.07554	0.07509	0.07491	**0.07466**	0.07468	0.07472
6	0.07673	0.07608	0.07546	0.07509	0.07485	0.07475	0.07467	0.07471
7	0.07671	0.07607	0.07556	0.07512	0.07486	0.07477	0.07476	0.07479
8	0.07677	0.07609	0.07557	0.07517	0.07483	0.07479	0.07475	0.07478

Table 2
Hyperparameter tuning results on the validation partition (MAE, m) for XGBoost along the dimensions of E = learning rate (eta), for a fixed optimal Max Tree Depth TD = 10 and Minimum Child Weight W = 5. Optimal settings shown in bold.

E	MAE
0.005	0.1368
0.010	0.0889
0.050	**0.0747**
0.100	0.0759
0.200	0.0783

of 0.074 m in H and 0.309 s in T_p, with maximum absolute errors of 0.853 m and 2.613 s respectively. This corresponds to MAPE values of 15.611% in H and 8.322% in T_p. The downwind station 45142 yielded larger errors, namely a MAE of 0.100 m in H and 0.516 s in T_p, with maximum absolute errors of 1.045 m and 3.774 s respectively. This corresponds to larger MAPE values of 22.890% in H and 13.445% in T_p. Scatterplot comparisons with other the models will be discussed in Section 6.

4. Long Short-Term Memory (LSTM) model

4.1. Model description

Our second model is a specialized class of artificial neural network, namely a Recurrent Neural Network (RNN). This class of network differs from the standard feed-forward type in that the time dependency of the input variables is explicitly modeled. RNN models are composed of a sequence of identical neural network cells, each one representing one time step. Each of these units passes its nonlinear activation to the unit

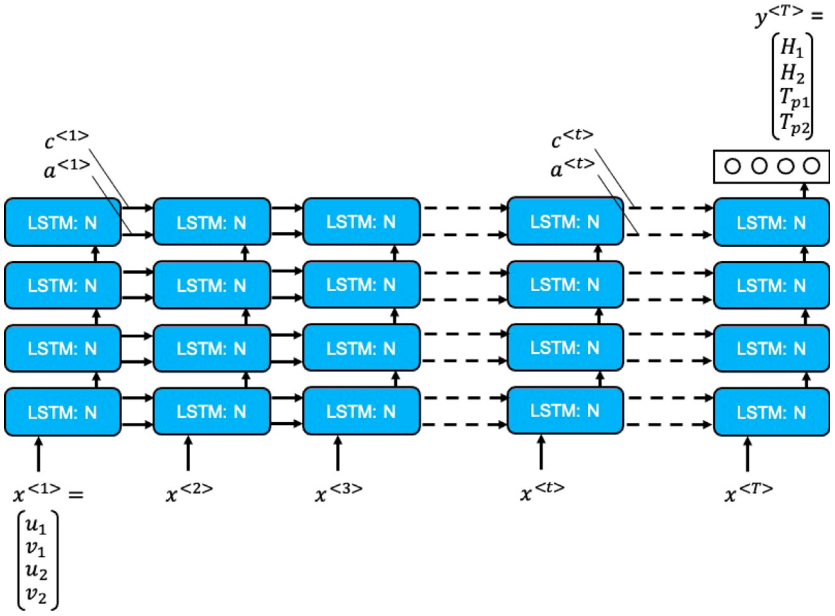


Fig. 10. Architecture of the 4-layer LSTM network.

of the next time step, thus allowing a system memory to build up. This system memory is useful for modeling sequences such as time series of physical phenomena and natural language. As motivated by the early work of Mandal and Prabaharan (2006), RNNs are therefore well-suited to the problem of wave height prediction, a phenomenon that is known to depend on a time series history of winds. Here we apply a popular variety of the RNN, namely the Long Short-Term Memory (LSTM) network by Hochreiter and Schmidhuber (1997). The structure of one cell of the LSTM is expressed in the following set of equations (see Olah (2015) and Chollet (2017) for graphical representations):

$$f^{\langle t\rangle} = \sigma\left(\mathbf{W}_f\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + \mathbf{b}_f\right) \tag{10}$$

$$i^{\langle t\rangle} = \sigma\left(\mathbf{W}_i\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + \mathbf{b}_i\right) \tag{11}$$

$$o^{\langle t\rangle} = \sigma\left(\mathbf{W}_o\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + \mathbf{b}_o\right) \tag{12}$$

$$\tilde{c}^{\langle t\rangle} = \tanh\left(\mathbf{W}_c\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + \mathbf{b}_c\right) \tag{13}$$

$$c^{\langle t\rangle} = i^{\langle t\rangle} * \tilde{c}^{\langle t\rangle} + f^{\langle t\rangle} * c^{\langle t-1\rangle} \tag{14}$$

$$a^{\langle t\rangle} = o^{\langle t\rangle} * \tanh\left(c^{\langle t\rangle}\right) \tag{15}$$

The input to the LSTM consists of a time series vector $\mathbf{x} = \left[x^{\langle 1\rangle}, x^{\langle 2\rangle}, \ldots, x^{\langle T\rangle}\right]$, for $t = 1 \ldots T$ time steps in the series. The vector $\mathbf{a} = \left[a^{\langle 1\rangle}, a^{\langle 2\rangle}, \ldots, a^{\langle T\rangle}\right]$ represents the network activation that is passed from one time step to the next, thus representing the short-term memory of the system. A second, longer-term memory is represented by the vector $\mathbf{c} = \left[c^{\langle 1\rangle}, c^{\langle 2\rangle}, \ldots, c^{\langle T\rangle}\right]$, which is also passed from one time step to the next. The distinction is that $\mathbf{c}$ (or the 'carry state') is not necessarily updated at every time step, unlike the activation $\mathbf{a}$. As such, $\mathbf{c}$ can carry information across several time steps, influencing model outputs based on inputs much earlier in the time series. The updating of $\mathbf{c}$ and its influence on the activation $\mathbf{a}$ is determined by a set of binary gates, called the 'forget gate' $f^{\langle t\rangle}$ given by (10), the 'update gate' $i^{\langle t\rangle}$ (11), and the 'output gate' $o^{\langle t\rangle}$ (12). In each of these expressions, σ is the sigmoid function, $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias vector. At each time step, a candidate $\tilde{c}^{\langle t\rangle}$ for updating the previous carry state $c^{\langle t-1\rangle}$ is computed using (13). In (14), based on the outcomes of the update and forget gates, $c^{\langle t\rangle}$ is then updated to the candidate $\tilde{c}^{\langle t\rangle}$ or just takes the same value as the previous carry state $c^{\langle t-1\rangle}$, thus continuing the long-term memory. Finally, the new activation $a^{\langle t\rangle}$ is computed based on the value of the output gate and the current carry state $c^{\langle t\rangle}$ using (15). The activation $a^{\langle t\rangle}$ is then used to determine the output $y^{\langle t\rangle}$ of the given time step, via e.g. a Rectified Linear Unit (ReLU) or softmax activation function. However, it is also passed forward to the next time step along with the carry state. This LSTM cell can thus be repeated for each time step in the sequence to be modeled, as will be shown in the next section.

To build our LSTM application in the Keras framework, the LSTM cell described in the previous section was arranged in a time sequence covering the 3-hourly time lags of wind observation going back to a maximum of 24 h, namely $t = [-24, -21, -18, -15, -12, -9, -6, -3, 0]$. This results in a horizontal sequence of up to 9 LSTM units for the wind input vectors $x^{\langle t\rangle}$ to $x^{\langle T\rangle}$. Each unit passes along its activation $a^{\langle t\rangle}$ and carry status $c^{\langle t\rangle}$ to the unit at the next time step (Fig. 10). The maximum number of lags T, which represents the duration of wave generation, will be determined via hyperparameter optimization below.

Each wind input vector $x^{\langle t\rangle} = [u_1, v_1, u_2, v_2]$ contains the wind velocity components at the two stations 45005 and 45142 at each time lag. These input sequences were generated by moving a time window through the data set, each time capturing a sequence of T wind input vectors. This results in a training data set with an input size of $[m_x, T, w] = [28\,154, T, 4]$, where m_x is the number of training sequences of length T and $w = 4$ the number of wind vector components. For the validation and test partitions the corresponding input sizes are $[5\,193, T, 4]$ and $[4\,493, T, 4]$, respectively. At each time step t, a series of four LSTM cells are stacked, each passing its activation to the next, to increase complexity of the nonlinear relationships that can be modeled. Each LSTM unit contains N neurons, which is the second hyperparameter to optimize. As described above, inputs are received at each time step $t = 1 \ldots T$. The output, however, is only generated at the last time step T, corresponding to the current time at which an estimate of the significant wave height is required. At this time step, the activation of the fourth LSTM layer is passed to a four-neuron dense layer with Rectified Linear Unit (ReLU) activations. This produces non-negative regression outputs of the significant wave heights and peak periods $y^{\langle T\rangle} = [H_1, H_2, T_{p1}, T_{p2}]$ at stations 45005 and 45142.

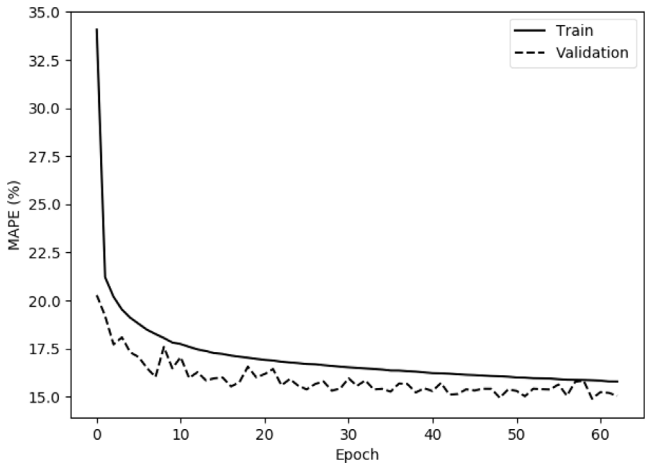


Fig. 11. Training performance of the LSTM network in terms of MAPE (%) of the training and validation partitions, for stations 45005 and 45142 combined.

4.2. Hyperparameter tuning

As described above, the LSTM structure features a sequence of N neurons per layer and T wind observations (Fig. 10). This neural network was trained using a mini-batch size of B, and RMSProp optimization with a learning rate of L. We therefore have four hyperparameters for this model, namely N, T, B, and L, which were determined using derivative-free optimization. Similar to the approach for the XGBoost model, a direct search strategy was followed: The search space was defined as $N = [12, 14, 16, 18, 20, 22, 24]$, $T = [3, 6, 9]$, $B = [16, 32, 64, 128]$ and $L = [0.001, 0.01, 0.1]$. For each hyperparameter combination, the LSTM model was trained for a maximum of 100 epochs, and the epoch at which the minimum validation partition MAPE was found was recorded. This minimum represents a balance between bias reduction and variance (overfitting) for that hyperparameter combination. Moving through the search space, the combination for which the minimum validation set MAPE was found is identified as the optimum set of hyperparameters. Tables 3 and 4 show the results of this optimization for different dimensions of the search space. Table 3 shows a preference in the optimization towards using an input wind history of 6 time steps, i.e. $t = [-15, -12, -9, -6, -3, 0]$, in particular when using slower learning rates. Furthermore, the slowest learning rate of $L = 0.001$ consistently yields the best results. Table 4 shows that the optimization is not sensitive to mini-batch size. This is because for a larger mini-batch size, the optimization responded by using more epochs to reach a minimum MAPE, and vice versa. An optimal combination of $N = 16$, $T = 6$, $B = 32$ and $L = 0.001$ was found, which is identified in bold in both tables. This model training ran on a CPU (Intel® Xeon® CPU E2176M @ 2.70 GHz) for about 3.5 min per hyperparameter combination.

Fig. 11 shows the training of this model in terms of aggregate MAPE over both stations 45005 and 45142 on the training and validation partitions, up to the optimal epoch. It can be seen that the error of both the training and validation partitions reduce rapidly during the first few epochs. Over the next decades of epochs the training partition score reduces steadily. The score of the validation set also reduces on average, although not as smoothly, considering the smaller sample size for the computation of these metrics. The MAPE of the validation set reaches a 3-point moving average minimum at 63 epochs, suggesting that the model is not overfit. After 63 epochs, the model achieves an aggregate MAPE = 15.995% over both stations for the training partition (years 1994–2013), an MAPE = 14.770% for the validation partition (years 2014–2015), and an MAPE = 15.328% for the unseen test partition (years 2016–2017). We can therefore conclude that the LSTM model is indeed not overfit, and seems to generalize well, yielding somewhat better results for the unseen test partition than the training and validation partition.

Table 3
Hyperparameter tuning results on the validation partition (MAPE) for LSTM along the dimensions of N = neurons per layer, T = number of lags, and L = learning rate, for a fixed optimal mini-batch size $B = 32$. Optimal settings shown in bold.

N	T	L		
		0.001	0.01	0.1
12	3	15.384	16.040	24.731
	6	14.895	15.873	25.302
	9	14.982	15.777	25.740
14	3	15.435	15.832	23.430
	6	14.864	15.746	26.915
	9	15.037	15.984	25.734
16	3	15.370	15.720	27.050
	6	**14.770**	15.914	27.906
	9	15.194	16.082	27.530
18	3	15.358	15.829	57.368
	6	14.962	16.074	40.982
	9	15.473	16.092	26.750
20	3	15.360	15.858	38.891
	6	14.968	15.943	54.723
	9	15.220	16.019	32.349
22	3	15.403	16.067	26.732
	6	15.014	15.857	39.388
	9	15.181	16.099	31.291
24	3	15.349	15.911	27.463
	6	14.937	15.933	33.704
	9	15.055	16.129	30.833

Table 4
Hyperparameter tuning results on the validation partition (MAPE) for LSTM along the dimensions of N = neurons per layer, T = number of lags, and B = mini-batch size, for a fixed optimal learning rate $L = 0.001$. Optimal settings shown in bold.

N	T	B			
		16	32	64	128
12	3	15.424	15.384	15.449	37.192
	6	14.851	14.895	14.994	14.866
	9	15.088	14.982	15.078	38.033
14	3	15.485	15.435	15.448	15.389
	6	14.878	14.864	15.028	14.854
	9	15.034	15.037	15.102	15.121
16	3	15.388	15.370	15.219	15.482
	6	14.989	**14.770**	14.985	14.863
	9	15.058	15.194	15.217	15.094
18	3	15.489	15.358	15.496	15.434
	6	15.056	14.962	15.008	14.917
	9	15.335	15.473	15.244	15.207
20	3	15.338	15.360	15.513	15.416
	6	15.133	14.968	14.946	14.898
	9	15.113	15.220	15.295	15.093
22	3	15.419	15.403	15.408	15.323
	6	15.077	15.014	38.123	14.776
	9	15.082	15.181	15.229	15.146
24	3	15.470	15.349	15.315	38.215
	6	14.971	14.937	36.862	14.932
	9	15.331	15.055	15.231	15.111

4.3. LSTM results

Figs. 12 and 13 show the predicted LSTM significant wave height and peak period, and their error statistics, for the unseen test partition (years 2016–2017). The LSTM model provides a good representation of the observed H and T_p in both the low- and high energy regimes. Importantly, the LSTM model captures all major wave events, including the storm peaks. At the upwind station 45005, the LSTM model yields an MAE of 0.103 m in H and 0.326 s in T_p, with maximum absolute

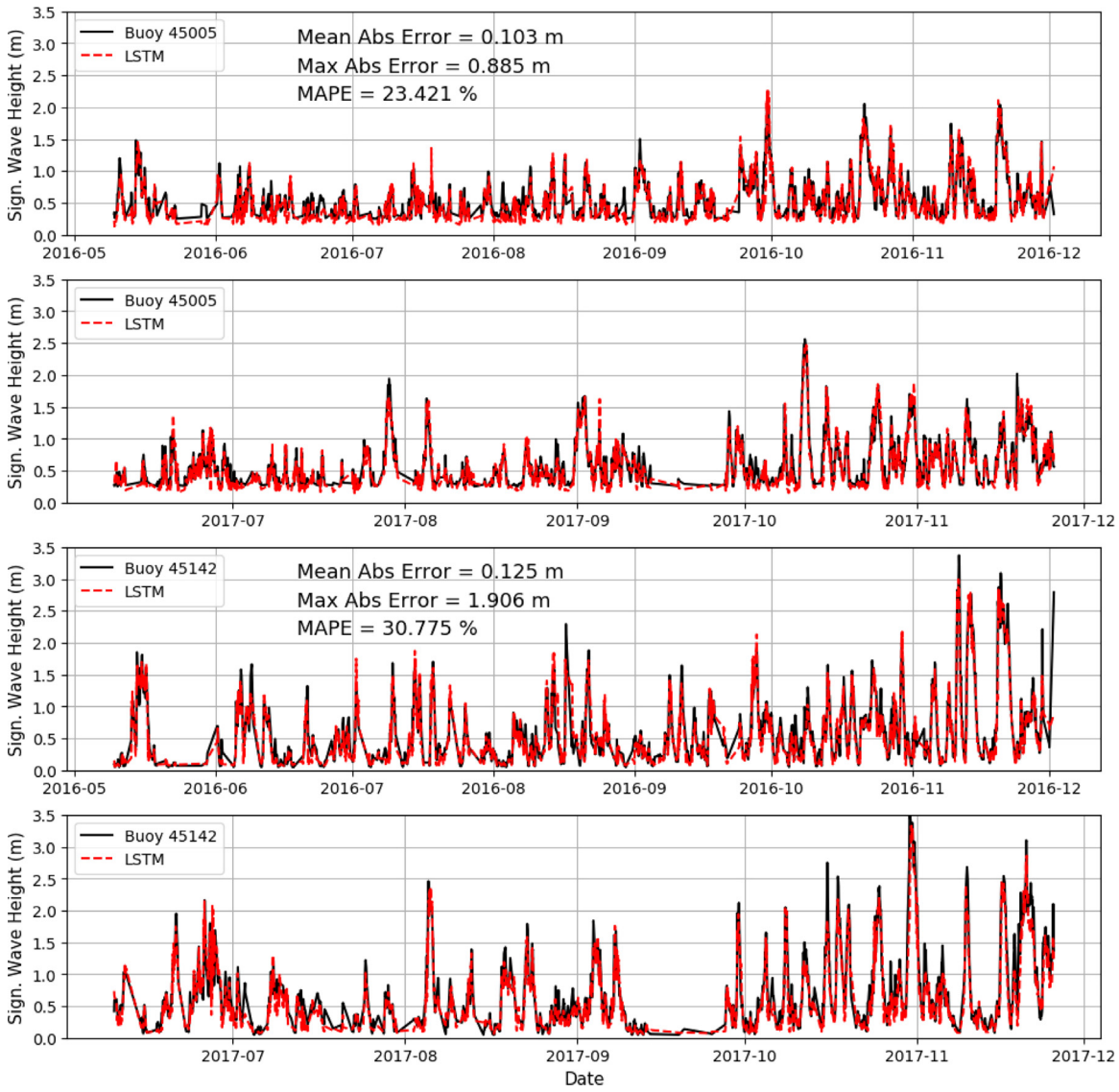


Fig. 12. Time series of the LSTM significant wave heights predictions and error statistics for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

errors of 0.885 m and 2.487 s respectively. This corresponds to MAPE values of 23.421% in H and 9.072 s in T_p. As was seen for the XGBoost model above, errors tend to be larger at the downwind station 45142: an MAE of 0.125 m in H and 0.499 s in T_p, with maximum absolute errors of 1.906 m and 3.831 s respectively. This corresponds to MAPE values of 30.775% in H and 13.614% in T_p. The trained LSTM needed only 0.24 s on 1 CPU to perform this 2-year prediction.

5. WAVEWATCH III spectral model

5.1. Model description

The unstructured mesh used for the WW3 runs is shown in Fig. 14, which was generated using Aquaveo's Surface-Water-Modeling System (SMS). The grid size distribution was configured as a function of the 3 arc-second bathymetry data from the National Centers for Environmental Information (NCEI). The model bathymetry was obtained by interpolating the observed bathymetry onto each unstructured grid node using the inverse distance method. High-resolution NOAA coastline data was applied to delineate the land boundary. The resulting mesh is composed of 11 509 triangular elements and 6 106 nodes in the horizontal. The resolution varies from approximately 100 m near the shore to 2.5 km offshore. The model has distribution referenced to the Great Lakes low water datum of 173.5 m.

Surface wind observations were interpolated to create hourly gridded surface meteorological analyses of wind. The observations are from the Great Lakes Weather Data and Marine Observations (GLERL, 2020), which includes coastal and offshore meteorological stations. These data were corrected for over-water conditions and interpolated, along with available in-lake buoys, to the model grid. Fig. 15 shows scatter and Quantile–Quantile (Q–Q) plots comparing the observed winds with the analysis winds used in WW3. From the Q–Q plots we can see that the analysis wind fields match the observation at 45005 well, although they slightly underestimate values greater than 10 m/s. Conversely, they slightly overestimate the observations at 45142 during strong events of around 15 m/s. We can therefore conclude that the wind fields applied to WW3 have a generally high accuracy, and only a limited bias.

The WW3 simulation starts at 12:00 GMT on 01 January 2016. Model results are output on an hourly interval at the same time step of the buoy data. The model took 24 h on 60 CPUs (Intel® Xeon® CPU E5-2603 v4 @1.70 GHz) to complete the prediction for 2016–2017. The model settings were chosen to match those of the current NOAA operational model for the Great Lakes (Alves et al., 2014), which are: Wind input and dissipation source terms based on Ardhuin et al. (2010).

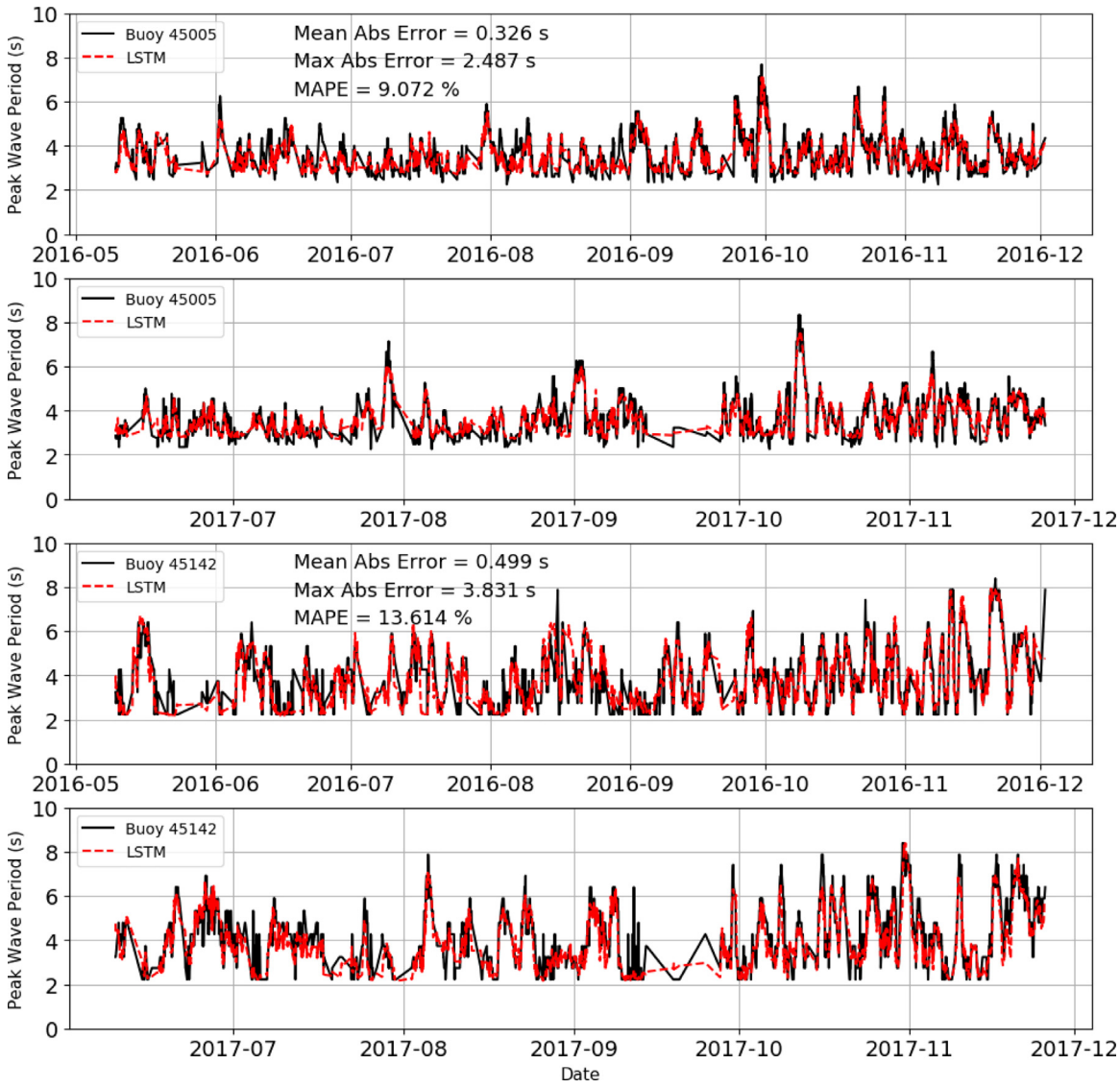


Fig. 13. Time series of the LSTM peak wave period predictions and error statistics for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

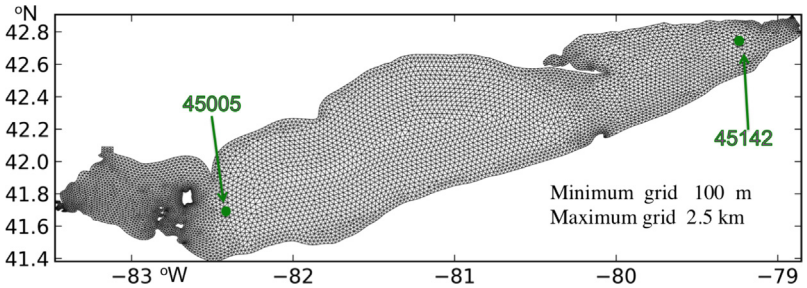


Fig. 14. Unstructured grid of the Lake Erie used for WW3. Green dots are the buoy stations of 45005 and 45142.

Flux computation included in the model source terms; The third-order propagation scheme with Tolman (2002) averaging technique; Generalized Multiple DIA (GMD) nonlinear interaction (Tolman, 2013); The empirical, linear JONSWAP parameterization of bottom friction (Hasselmann et al., 1973); Depth-induced breaking according to Battjes and Janssen (1978); Linear time interpolation for wind interpolation, and approximately linear speed interpolation for wind space interpolation; No triad interactions used; No bottom scattering used; No supplemental source term used. The model was compiled with NetCDF version 4 as a stand-alone program. The model used Message Passing Interface (MPI) with a distributed memory. For more details, refer to Alves et al. (2014). We used this model configuration as a benchmark, and no additional tuning was done.

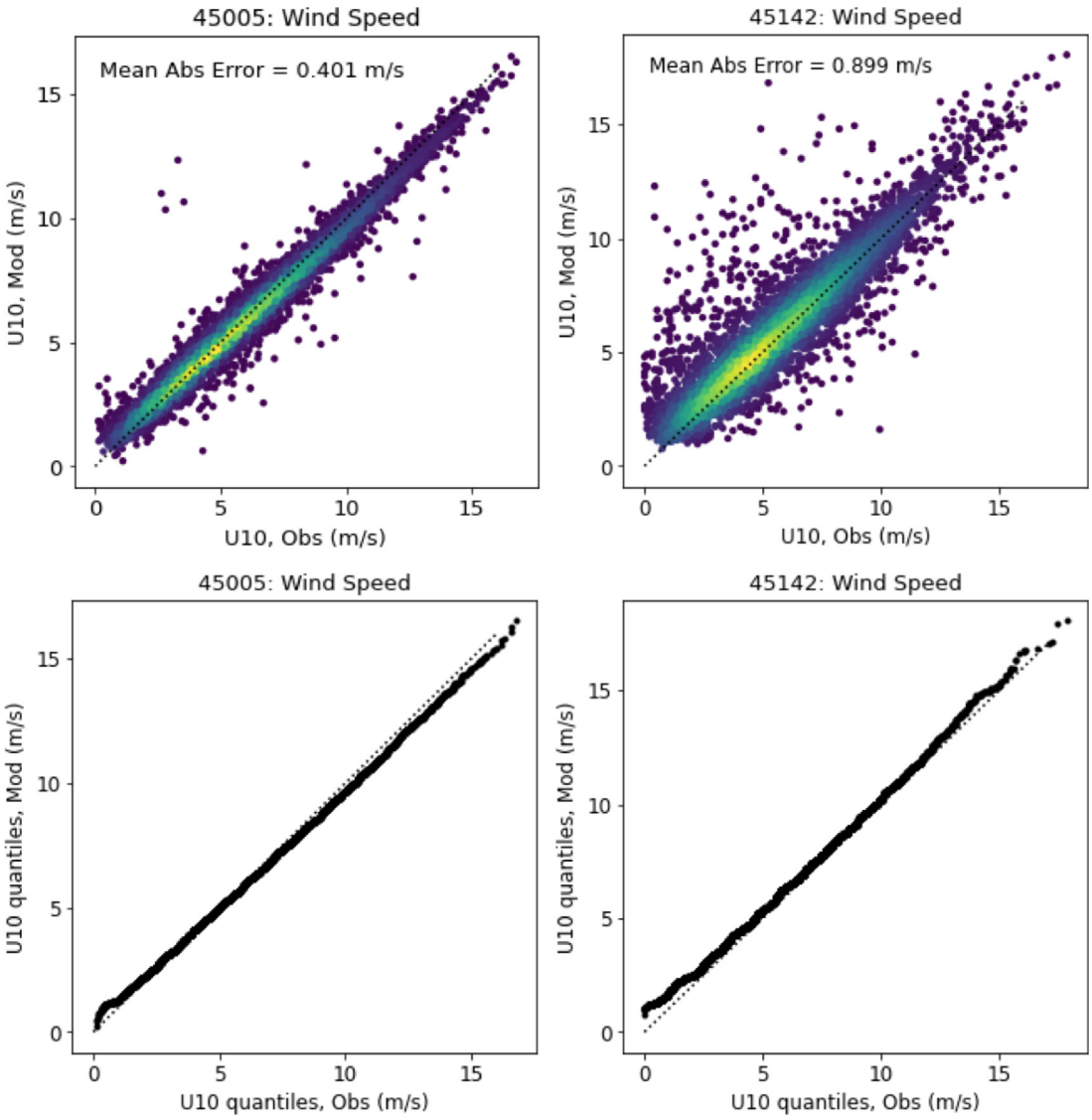


Fig. 15. Top: Scatterplots of U_{10} wind used for WW3 against wind observations at NDBC 45005 (left) and 45142 (right) for the test partition (2016–2017). Colors indicate the relative data density. Bottom: Q–Q plots of the same data.

5.2. WW3 results

The WW3 model was implemented to predict significant wave height and peak wave period for 2016–2017. Fig. 16 shows the time series of modeled and observed significant wave height. For station 45005, a MAE of 0.082 m and a maximum absolute error of 0.873 m were found. For station 45142, the model yielded an MAE of 0.119 m, and a maximum absolute error of 1.100 m. This corresponds to MAPE values of 15.325% and 21.008% for the two stations respectively. Fig. 17 shows the corresponding results for peak wave period. At station 45005, a MAE of 0.408 s and a maximum absolute error of 2.604 s were found, and at station 45142 a MAE of 0.653 s and a maximum absolute error of 5.864 s were found. The MAPE values are 12.506% and 19.318% for the two stations respectively. These figures show that WW3 tends to underestimate wave height and period peaks during strong wind events (spikes on the figures). We note that the slightly underestimated model winds at station 45005 during strong wind events (Fig. 15, left) may be the cause of the underestimation of these peaks. However, this cannot explain the behavior at the downwind station 45142, where the model winds slightly overestimate observations during strong wind events (Fig. 15, right), while the wave height and period peaks are still underestimated. We tested an alternative wind data set from an operational weather forecast model by the High Resolution Rapid Refresh (HRRR, Benjamin et al., 2016a,b) in WW3 simulations, and the issue with underestimating the peak significant wave height still persisted and the maximum absolute errors were slightly worse than the results with the interpolated winds from the observations.

6. Model intercomparison

6.1. Significant wave height

Fig. 18 compares the performance of the two ML models and WW3 in terms of scatter plots of significant wave height at stations 45005 and 45142 for the test partition (2016–2017). These results are summarized in Table 5. In addition to the MAPE and MAE discussed above, we also consider the Scatter Index (SI), Correlation Coefficient (CC), bias, relative bias, and regression fit.

The XGBoost model (Fig. 18, top row) displays a tight grouping around the line of perfect agreement, in particular in the lower range of wave heights, where the data density is the highest (warmer colors). We see a low scatter (MAE = 0.074 m; MAPE = 15.61%; SI = 0.160; CC = 0.958) and small overall bias (−0.011 m or −1.81%) against the

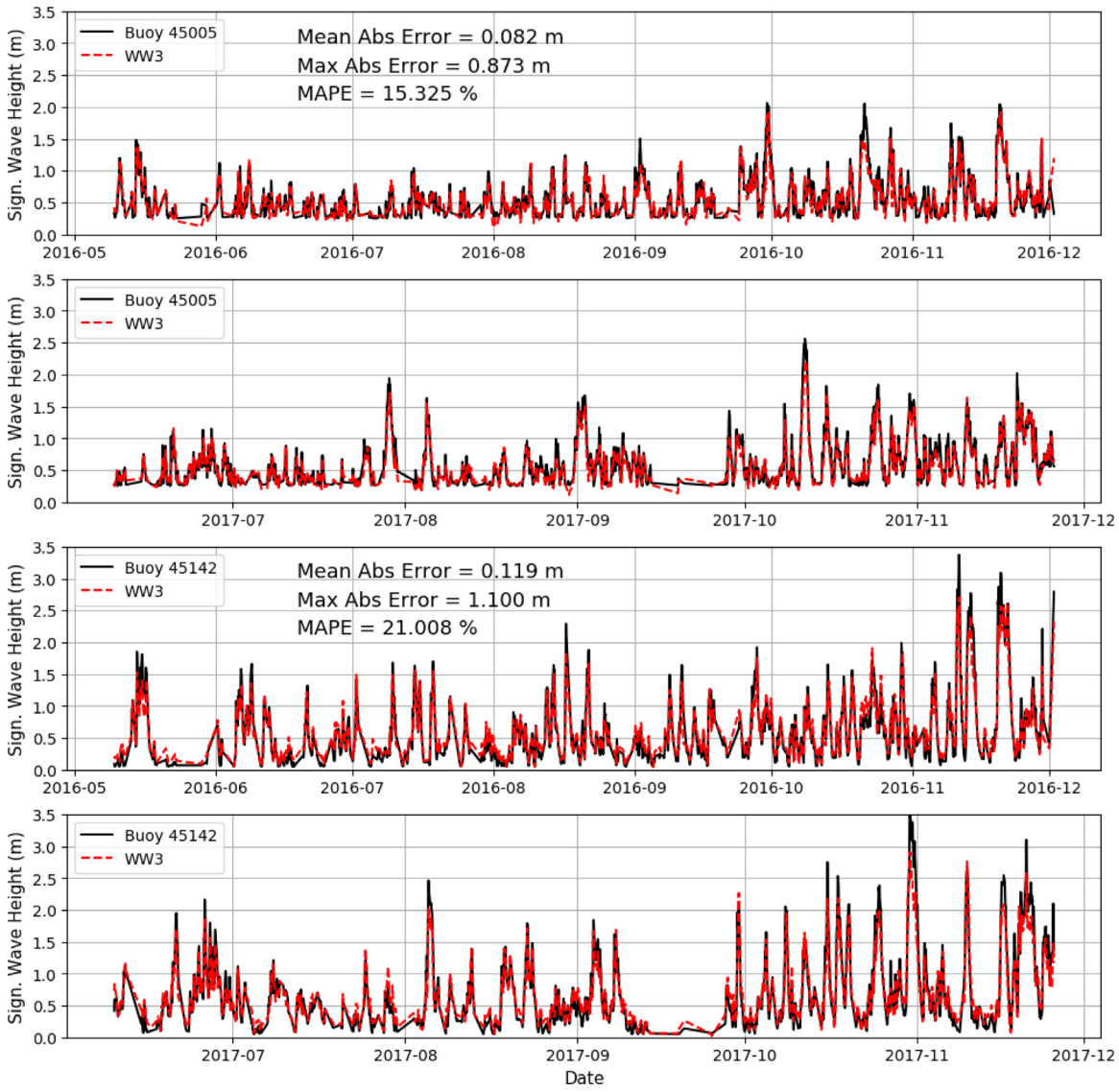


Fig. 16. Time series of the WW3 significant wave heights predictions and error statistics for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

observations at the upwind station 45005. To assess the performance during storm conditions, the bias is also computed over observations greater than the 95th percentile (diamond markers, score included in parentheses), which also displays a small bias (−0.042 m or −2.56%). The regression fit through the results indicate a slope close to unity (0.974), and an intercept of close to zero (0.005 m). At the downwind station 45 142, the scatter increases (MAE = 0.100 m; MAPE = 21.89%; SI = 0.220; CC = 0.970), and the overall bias increases to −0.063 m (−9.28%). We see a comparable bias of −0.239 m (−10.61%) over the data points greater than the 95th percentile. This underprediction of H is reflected in the lower regression slope of 0.895. We note that the larger errors at the downwind station 45142 compared to the upwind 45005 is presumably due to the added complexity of its longer fetch over this elongated lake, which is a nonlinear function of the wind direction (refer Figs. 2 and 7).

The LSTM model (Fig. 18, center row) shows similar, but weaker performance than the XGBoost model in terms of significant wave height. At the upwind station 45005, the scatter is larger (MAE = 0.103 m; MAPE = 23.42%; SI = 0.221; CC = 0.931), and the overall bias (−0.037 m or −5.91%) and the bias of values greater than the 95th percentile (−0.054 m or −3.34%) are somewhat higher. The regression slope is close to unity (0.989), and the intercept close to zero (−0.030 m). At the downwind station 45142, the scatter increases relative to XGBoost and WW3 (MAE = 0.125 m; MAPE 30.78%; SI = 0.269; CC = 0.948), and the overall bias increases to −0.059 m (−8.71%). However, for values greater than the 95th percentile, the negative bias is the lowest of the three models at −0.201 m (−8.93%). This underestimation is reflected in the regression slope of 0.909, which is nonetheless the closest to unity of the three models.

The performance of WW3 (Fig. 18, bottom row) in terms of scatter is slightly poorer than XGBoost and better than LSTM, both at station 45005 (MAE = 0.082 m; MAPE = 15.33%; SI = 0.180; CC = 0.950) and 45142 (MAE = 0.119 m; MAPE = 21.01%; SI = 0.241; CC = 0.963). In terms of overall bias, WW3 performs similar or better than the two ML models, with values of −0.024 m (−3.91%) and −0.002 m (−0.33%) at stations 45005 and 45142 respectively. However, these overall bias scores obscure a pattern of underestimation in the upper range of wave heights. The bias values over values greater than the 95th percentile are significantly negative at −0.191 m (−11.76%) and −0.318 m (−14.15%), at stations 45005 and 45142 respectively. This underestimation is reflected in the poorer regression slopes of 0.834 and 0.769 for these stations respectively. Conversely, an overestimation in the lower ranges can be seen from the regression intercepts, which at 0.079 m and 0.135 m indicates a marked deviation from zero.

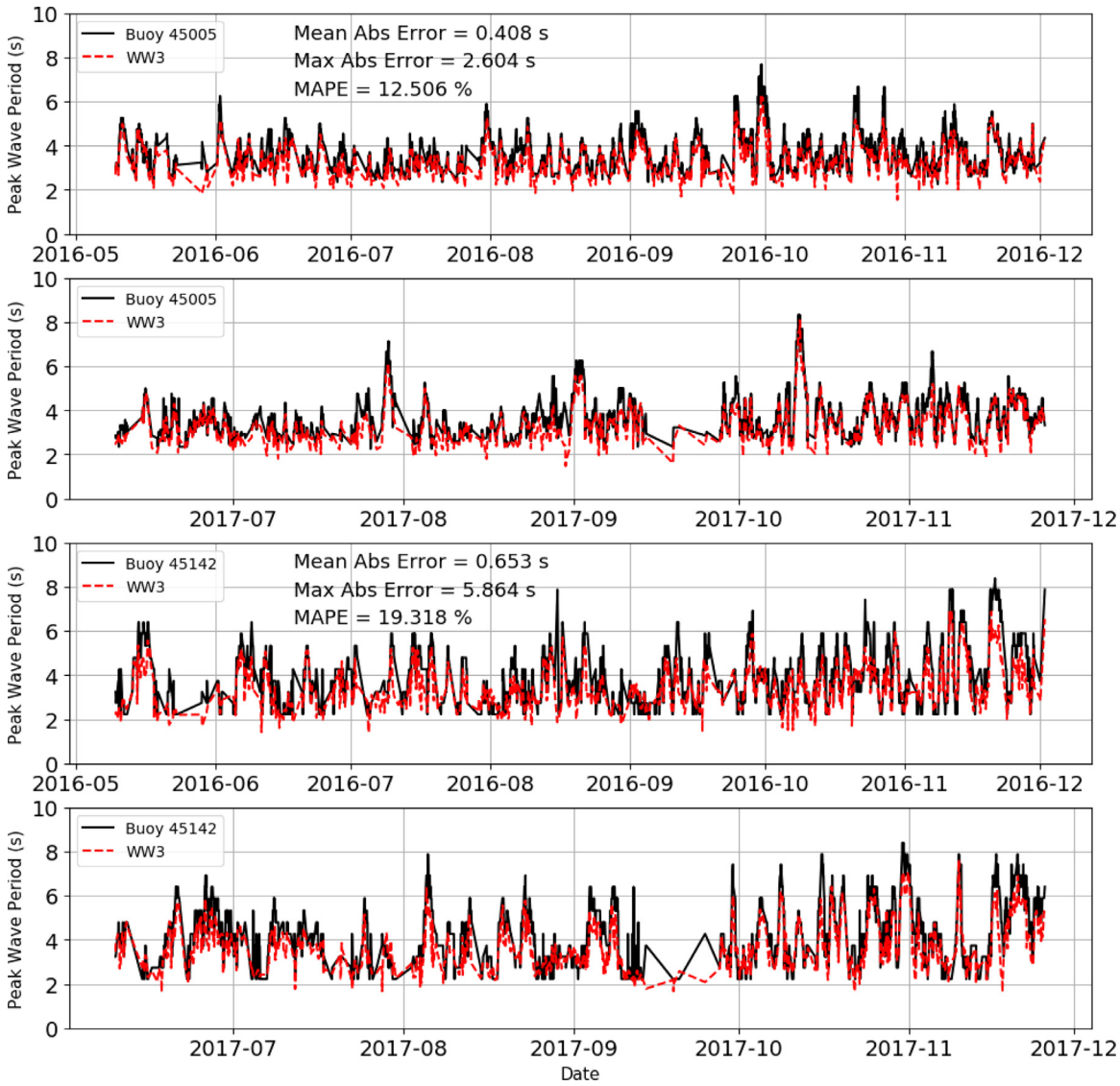


Fig. 17. Time series of the WW3 peak wave period predictions and error statistics for the unseen test partition (2016–2017). Top panels: NDBC 45005; Bottom panels: NDBC 45142.

Fig. 19 presents the corresponding Q–Q plots in order to compare the distributions of the simulated significant wave heights to those of the observations. In particular, the upper quantiles (right-hand tail of each distribution) represent the behavior of the modeled storm peaks. The distributions of the XGBoost and LSTM model at stations 45005 and 45142 follow those of the observations closely. However, some negative bias can be seen towards higher values at 45142, reflecting a tendency to underestimate storm peaks. This result might be surprising, considering that ML models such as neural networks are typically unbiased when properly trained. The finding that the lower quantiles agree better with observations in these ML models can be explained by the fact that the majority of the available training data cover this lower significant wave height range (75th percentile = 0.8 m, see Fig. 7). This concentration of data within the lower wave height range can also be seen from the color mapping in Fig. 18. Improved model fitting at these upper quantiles can be obtained by including more storm peak observations in the training set, or alternatively by using weighted samples with greater weights at these higher values (not shown). Furthermore, Mahjoobi and Etemad-Shahidi (2008), Prahlada and Deka (2015) and Dixit and Londhe (2016) suggest dividing the data into multiple parts, and training a separate model on the higher wave height partition. This is conceptually similar to threshold autoregressive models (TAR) in time series statistics (e.g. Shumway and Stoffer, 2017).

By comparison, the quantiles of the WW3 model have a clear low bias compared to observations, which becomes progressively larger with increasing significant wave heights. This indicates that the WW3 results are left-skewed compared to the distribution of the observed wave heights, i.e. that the extremes (storm peaks) are underpredicted by the model. Cavaleri (2009) has identified the underprediction of storm peaks as the most challenging aspect of physics-based wave models. Due to the difficulty of observing detailed wave dynamics under extreme conditions, the theories and empirical expressions incorporated into these models are often based on moderate storm conditions, and may not include potential nonlinear effects occurring during extreme events. As a result, these models can have difficulty in generalizing to such conditions.

6.2. Peak wave period

Fig. 20 compares the performance of the two ML models and WW3 in terms of scatter plots of peak wave period at stations 45005 and 45142 for the test partition (2016–2017). These results are summarized in Table 6.

The XGBoost model (Fig. 20, top row) yields the lowest scatter of the three models considered, although values do increase moving from

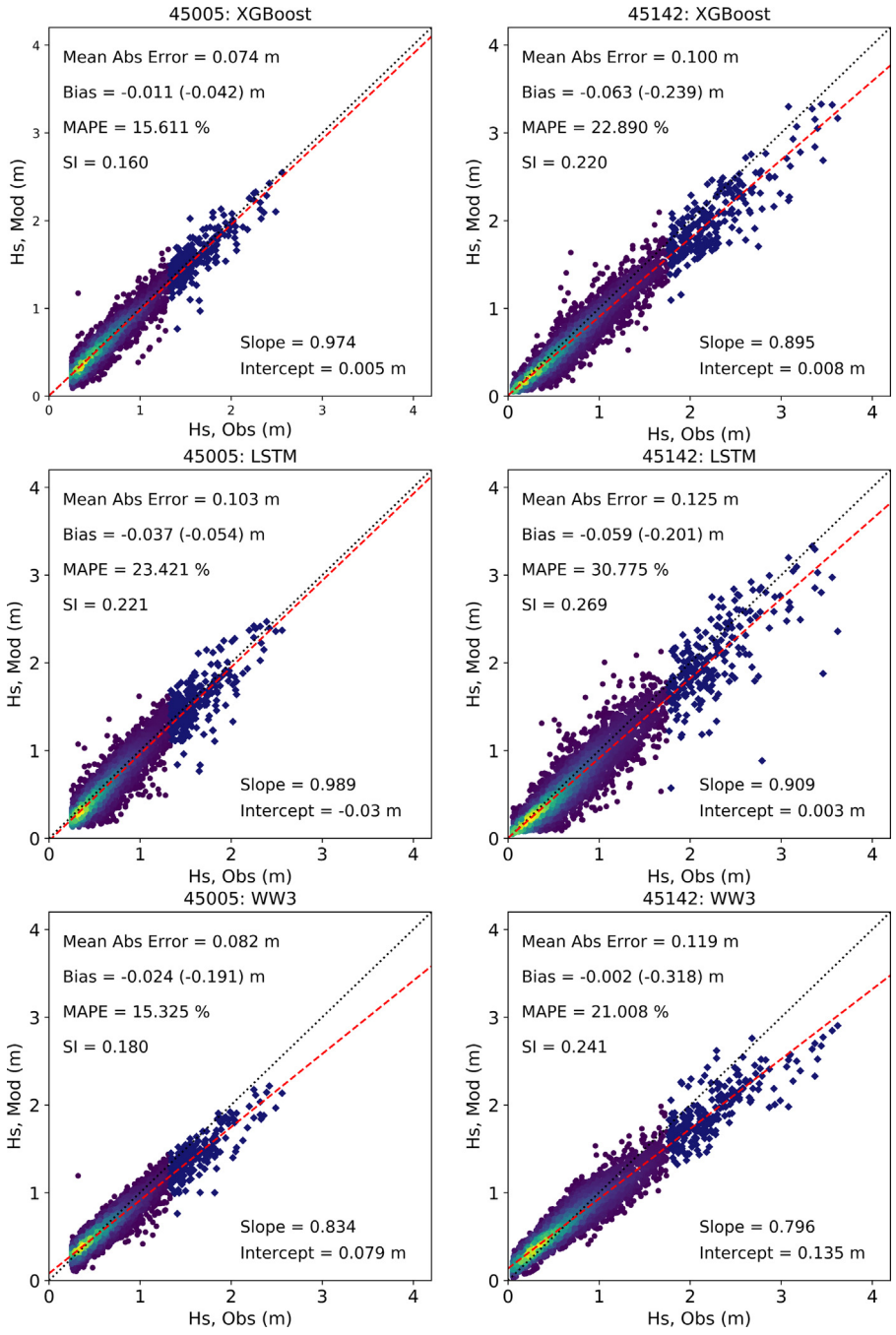


Fig. 18. Scatterplots of significant wave height for XGBoost (top), LSTM (center), and WW3 (bottom) against observations at NDBC 45005 (left) and 45142 (right) for the test partition (years 2016–2017). Colors indicate the relative data density. The line of perfect agreement in black dotted. Linear regression of observations vs model in red dashed. Bias value in parentheses computed over data greater than the 95th percentile (diamond markers).

the upwind station 45005 (MAE = 0.309 s; MAPE = 8.32%; SI = 0.116; CC = 0.866) to the downwind station 45142 (MAE = 0.516 s; MAPE = 13.45%; SI = 0.172; CC = 0.835). Overall bias levels remain equally low between the two stations, at −0.033 s (−0.87%) and −0.061 s (−1.47%) respectively, although they become more prominently negative for values greater than the 95th percentile (parentheses), at −0.574 s (−9.62%) and −0.690 s (−10.13%) respectively. As a result, the regression slopes are relatively low at 0.730 and 0.707 for stations 45005 and 45142 respectively.

The LSTM model (Fig. 20, center row) yields scatter results that are somewhat larger than those of XGBoost, again increasing from station 45005 (MAE = 0.326 s; MAPE = 9.07%; SI = 0.120; CC = 0.860) to the downwind station 45 142 (MAE = 0.499 s; MAPE = 13.61%; SI = 0.166; CC = 0.859). However, the LSTM model captures the peak periods of storm waves better, so that the biases over values greater than the 95th percentile are the lowest of the three models, with −0.482 s (−8.08%) and −0.538 s (−7.89%) at stations 45005 and 45142 respectively. This is reflected in the regression slopes of 0.758 and 0.826 respectively, which are the best of the three models.

By comparison, the WW3 model (Fig. 20, bottom row) yields the poorest results on all metrics considered. The scatter results are larger than both the XGBoost and LSTM, both at station 45005 (MAE = 0.408 s; MAPE = 12.51%; SI = 0.147; CC = 0.865) and 45142 (MAE = 0.653 s; MAPE = 19.32%; SI = 0.216; CC = 0.812). WW3 displays overall

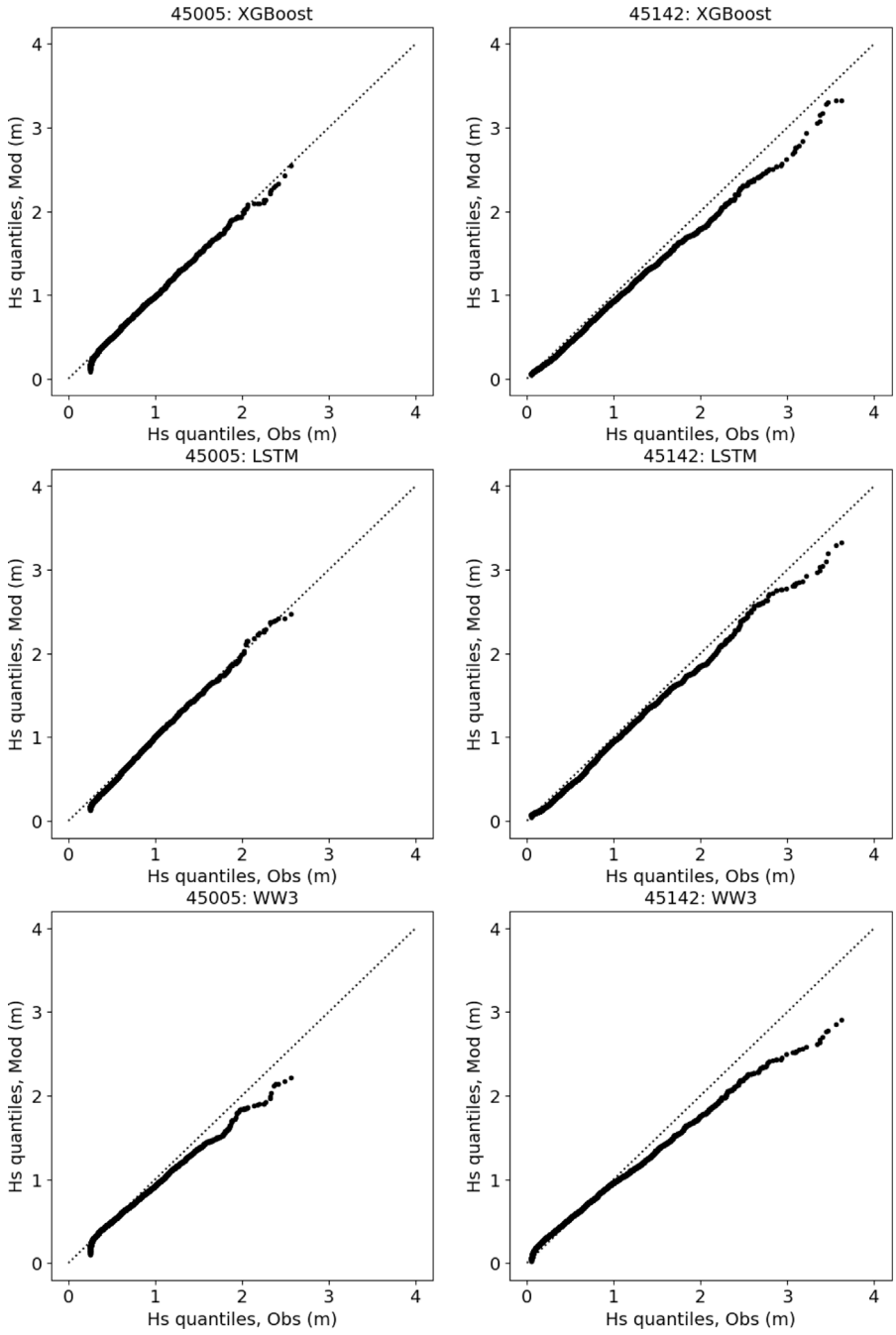


Fig. 19. Q–Q plots of observed and modeled significant wave height for the test partition (years 2016–2017). The models are XGBoost (top), LSTM (center), and WW3 (bottom) at NDBC 45005 (left) and 45142 (right).

negative biases of −0.339 s (−9.02%) and −0.474 s (−11.51%) at 45005 and 45142 respectively, with particularly large values of −0.929 s (−15.59%) and −1.342 s (19.68%), respectively, for values greater than the 95th percentile. This results in low regression slope values at station 45005 (0.744) and particularly at station 45142 (0.626).

This general underprediction of peak wave periods by WW3 is highlighted in the Q–Q plots (Fig. 21), where its quantiles can be seen to be below the line of perfect agreement in both the mid- and upper ranges. By contrast, the XGBoost and LSTM models display good distributions throughout. It can thus be concluded that these two ML models yield a clear improvement in the prediction of peak wave period at the two stations considered.

6.3. Model ensemble

In the model comparisons above, we have seen an overall best performance of the XGBoost compared to the other two models. However, we have also seen that the LSTM yielded the best performance in terms of values greater that the 95th percentile (storm peaks) in both significant wave height and peak wave period. Therefore, following Pirhooshyaran et al. (2020) we investigate the application of an ensemble of these three models, in order to assess if their combined performance can improve on those of the individual members.

Fig. 22 and the last two columns of Tables 5 and 6 show the results of the ensemble mean of the three component models. The top row of Fig. 22 shows that the ensemble mean has a low scatter and low

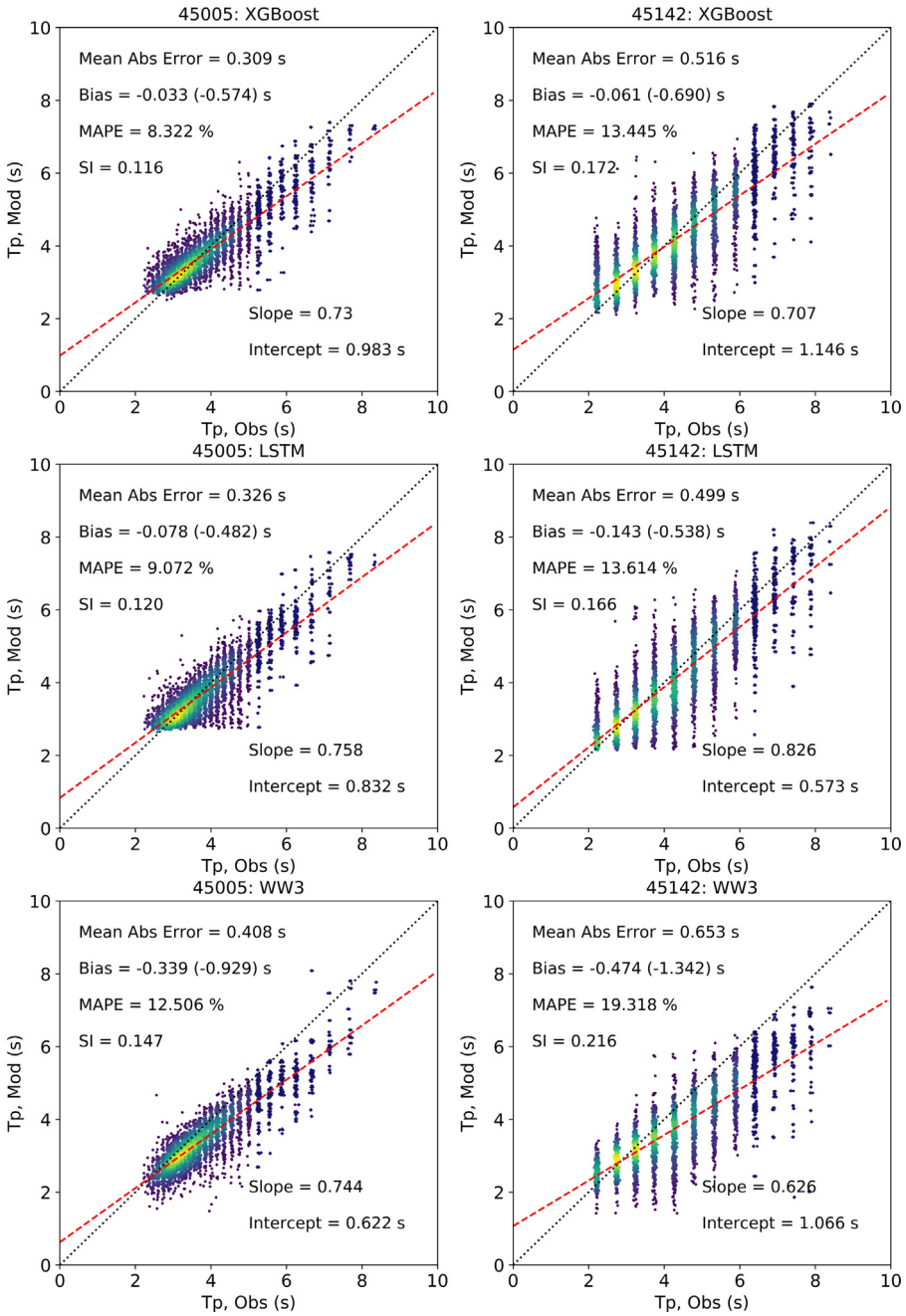


Fig. 20. Scatterplots of peak wave period for XGBoost (top), LSTM (center), and WW3 (bottom) against observations at NDBC 45005 (left) and 45142 (right) for the test partition (years 2016–2017). Colors indicate the relative data density, and observations were plotted with a small random variation (jitter) to distinguish individual data points. The line of perfect agreement in black dotted. Linear regression of observations vs model in red dashed. Bias value in parentheses computed over data greater than the 95th percentile (diamond markers).

overall bias. The scatter values at stations 45005 (MAE = 0.068 m; MAPE = 13.33%; SI = 0.151; CC = 0.964) and 45142 (MAE = 0.090 m; MAPE = 17.27%; SI = 0.195; CC = 0.977) are the lowest of the three component models considered (see also Table 5). By comparison, the bias results, in particular those of the values greater that the 95th percentile (parentheses), are an improvement over those of WW3, but they fall short of the improvements seen in the XGBoost and LSTM members. A similar outcome is seen for the peak wave period (Fig. 22, bottom row). The scatter results of the ensemble mean improve on those of the individual members, both at station 45005 (MAE = 0.287 s; MAPE = 7.96%; SI = 0.108; CC = 0.904) and station 45142 (MAE = 0.482 s; MAPE = 12.98%; SI = 0.157; CC = 0.885). However, the ensemble mean bias (both overall and values greater that the 95th percentile), performs better than WW3, but worse than the stronger ensemble members (LSTM and XGBoost).

The Q–Q plots in Fig. 23 confirm these bias results of the ensemble mean. For the significant wave height (top row), we see that the distribution has improved relative to that of WW3 (compare Fig. 19, bottom row), but still represents an underestimation relative to those of the LSTM and XGBoost (Fig. 19, center and top rows). Similarly, for the peak wave period (Fig. 23, bottom row), the ensemble mean improves on the results of WW3 (compare Fig. 21, bottom row), but performs poorer than the LSTM and XGBoost component models (Fig. 21, center and top rows). We can therefore conclude that the ensemble mean generally improves the component model performance in terms of scatter, but does not yield the best results in terms of correcting the negative bias during storm peaks.

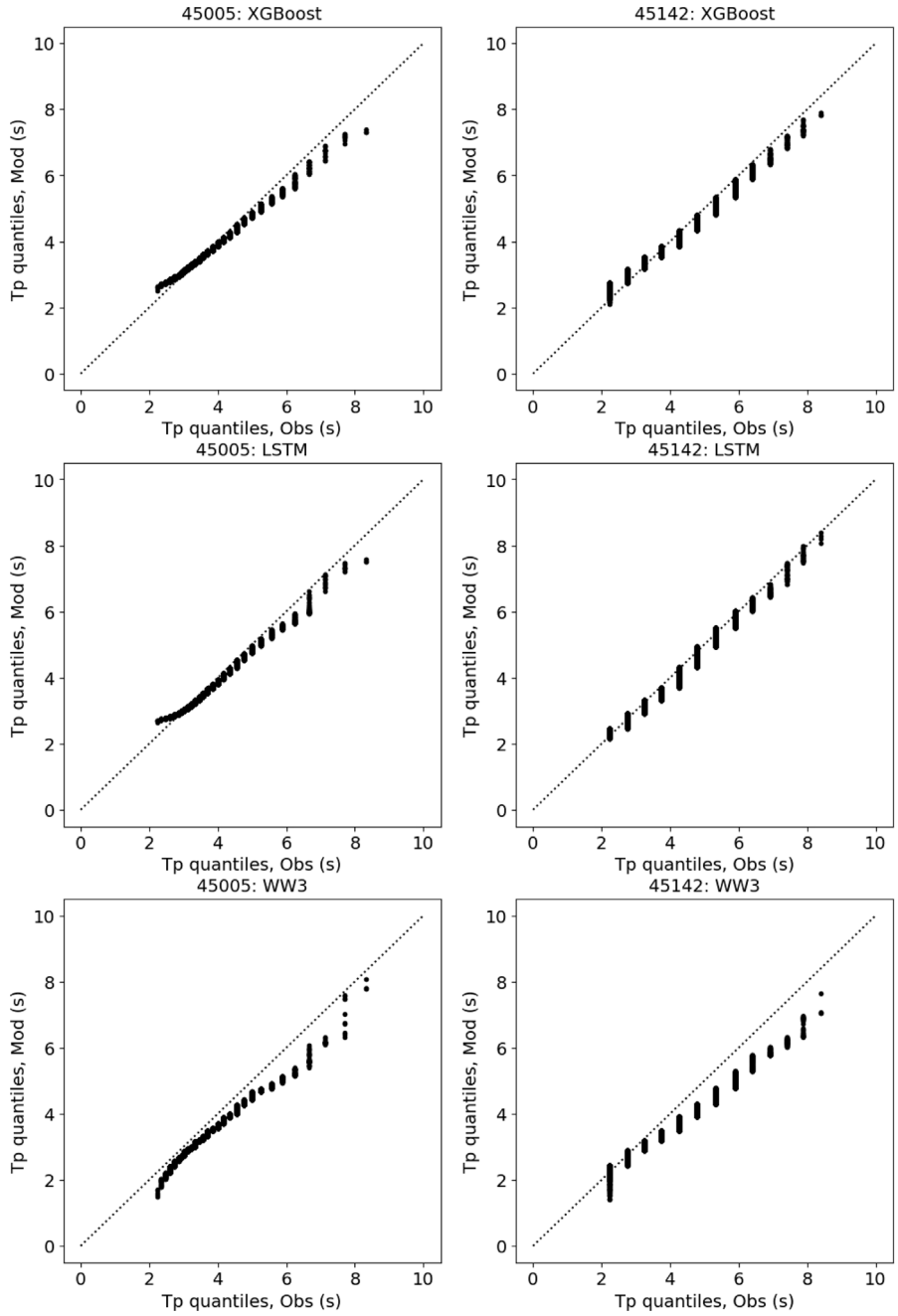


Fig. 21. Q–Q plots of observed and modeled peak wave period for the test partition (years 2016–2017). The models are XGBoost (top), LSTM (center), and WW3 (bottom) at NDBC 45005 (left) and 45142 (right).

7. Model sensitivity

In the previous section, we showed that advanced ML models can be effective in predicting significant wave height and peak wave period at point locations. However, it is well-known that such data-driven models are limited in their generality to the data sets on which they were trained. Here we investigate the models' sensitivity to data set size.

To investigate the effect of data set size, we conducted another training for the XGBoost and LSTM models in which we limited the training partition size to only 5 years (2011–2015), using the trained models to predict significant wave height for the same test period as before (2016–2017). With this retrained XGBoost model, the MAE for station 45005 is 0.080 m, with a maximum absolute error of 0.910 m. For station 45142, the MAE is 0.101 m with a maximum absolute error of 1.036 m. Fig. 24 shows that the original model trained with 15 years of data performs somewhat better than the one trained with only 5 years of data. However, the performance of the model fit on the longer training period is not much better, which means that it is close to saturation on the training data after about 5 years. This is typical for a tree-based model—it builds up enough tree representations when it sees enough samples, most importantly, the largest and the smallest.

For comparison, Fig. 25 shows the results of the LSTM model that was trained on the shorter period of 5 years. With this retrained model, the MAE for station 45005 is 0.090 m, with a maximum absolute error of 0.887 m. For station 45142, the MAE is 0.147 m with a maximum absolute error of 1.764 m. These MAE results are better than those of the LSTM trained on the 15-year data set for the upwind station 45005, but worse for the more geographically complex downwind station

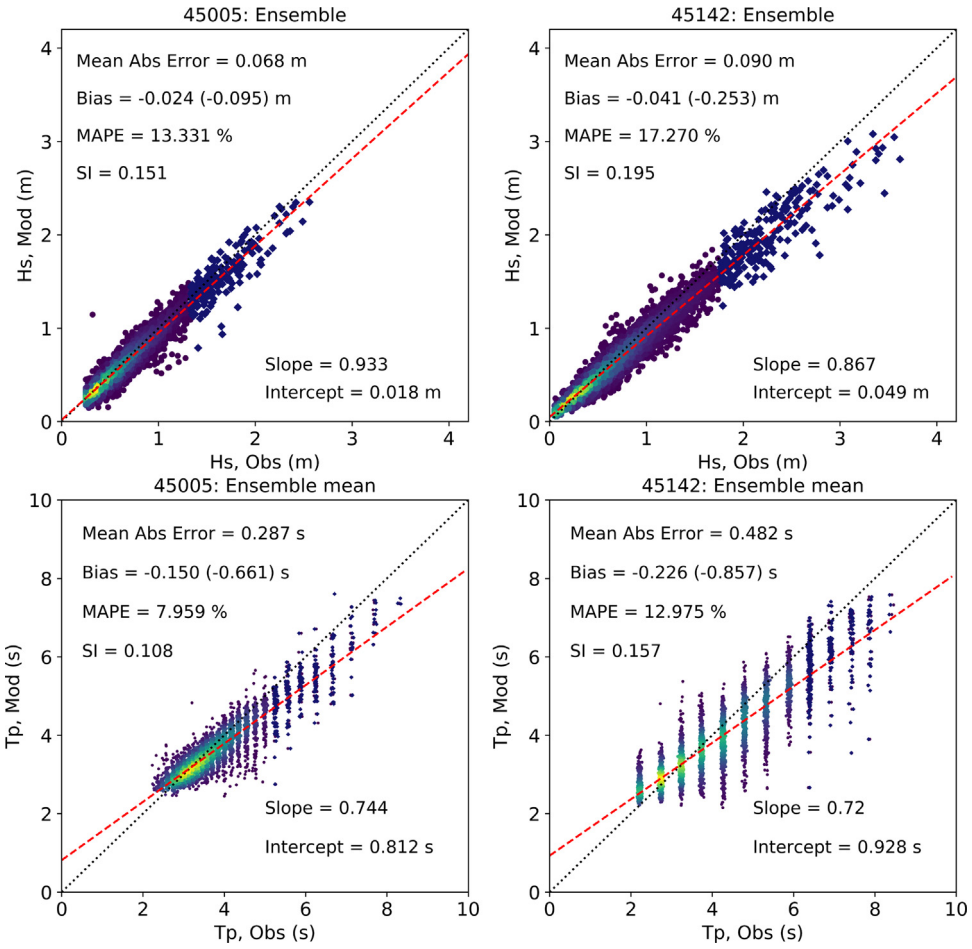


Fig. 22. Scatterplots of ensemble mean significant wave height (top) and peak wave period (bottom) against observations at NDBC 45005 (left) and 45142 (right) for the test partition (years 2016–2017). Colors indicate the relative data density, and peak wave period observations were plotted with a small random variation (jitter) to distinguish individual data points. The line of perfect agreement in black dotted. Linear regression of observations vs model in red dashed. Bias value in parentheses computed over data greater than the 95th percentile (diamond markers).

Table 5
Comparison of model performance for significant wave height at the buoy stations 45005 and 45142. Included are: Mean Absolute Error (MAE), maximum absolute error (Max_AE), Mean Absolute Percentage Error (MAPE), Scatter Index (SI), bias, bias >95th percentile (Bias_95), relative bias (in %), relative bias >95th percentile (Bias_95, in %), slope, intercept, and Correlation Coefficient (CC). Best result per metric and buoy shown in bold.

Metric	XGBoost		LSTM		WW3		Ens mean	
	45005	45142	45005	45142	45005	45142	45005	45142
MAE (m)	0.074	0.100	0.103	0.125	0.082	0.119	**0.068**	**0.090**
Max_AE (m)	0.853	1.045	0.885	1.906	0.873	1.100	**0.826**	**1.015**
MAPE (%)	15.611	22.890	23.421	30.775	15.325	21.008	**13.331**	**17.270**
SI (–)	0.160	0.220	0.221	0.269	0.180	0.241	**0.151**	**0.195**
Bias (m)	**−0.011**	−0.063	−0.037	−0.059	−0.024	**−0.002**	−0.024	−0.041
Bias_95 (m)	**−0.042**	−0.239	−0.054	**−0.201**	−0.191	−0.318	−0.095	−0.253
Bias (%)	**−1.81**	−9.28	−5.91	−8.71	−3.91	**−0.33**	−3.88	−6.11
Bias_95 (%)	**−2.56**	−10.61	−3.34	**−8.93**	−11.76	−14.15	−5.89	−11.23
Slope (–)	0.974	0.895	**0.989**	**0.909**	0.834	0.796	0.933	0.867
Intercept (m)	**0.005**	0.008	−0.030	**0.003**	0.079	0.135	0.018	0.049
CC (–)	0.958	0.970	0.931	0.948	0.950	0.963	**0.964**	**0.977**

Table 6
Comparison of model performance for peak wave period at the buoy stations 45005 and 45142. Included are: Mean Absolute Error (MAE), maximum absolute error (Max_AE), Mean Absolute Percentage Error (MAPE), Scatter Index (SI), bias, bias >95th percentile (Bias_95), relative bias (in %), relative bias >95th percentile (Bias_95, in %), slope, intercept, and Correlation Coefficient (CC). Best result per metric and buoy shown in bold.

Metric	XGBoost		LSTM		WW3		Ens mean	
	45005	45142	45005	45142	45005	45142	45005	45142
MAE (s)	0.309	0.516	0.326	0.499	0.408	0.653	**0.287**	**0.482**
Max_AE (s)	2.613	**3.774**	**2.487**	3.831	2.604	5.864	2.510	3.981
MAPE (%)	8.322	13.445	9.072	13.614	12.506	19.318	**7.959**	**12.975**
SI (–)	0.116	0.172	0.120	0.166	0.147	0.216	**0.108**	**0.157**
Bias (s)	**−0.033**	**−0.061**	−0.078	−0.143	−0.339	−0.474	−0.150	−0.226
Bias_95 (s)	−0.574	−0.690	**−0.482**	**−0.538**	−0.929	−1.342	−0.661	−0.857
Bias (%)	**−0.87**	**−1.47**	−2.06	−3.47	−9.02	−11.51	−3.98	−5.48
Bias_95 (%)	−9.62	−10.13	**−8.08**	**−7.89**	−15.59	−19.68	−11.10	−12.57
Slope (–)	0.730	0.707	**0.758**	**0.826**	0.744	0.626	0.744	0.720
Intercept (s)	0.983	1.146	0.832	**0.573**	**0.622**	1.066	0.812	0.928
CC (–)	0.866	0.835	0.860	0.859	0.865	0.812	**0.904**	**0.885**

45142. We can therefore conclude that unlike the XGBoost model, the additional data used in the original LSTM training did improve the quality of the model. Thus, XGBoost would be a better choice for this wave modeling application if only a few years of training data would be available.

8. Conclusions

In this study, ML methods based on XGBoost and LSTM were applied to wind-wave prediction in Lake Erie, and were compared to the physics-based spectral wave model WW3. This enclosed basin, with a wind climate directed along its major axis, represents near-idealized

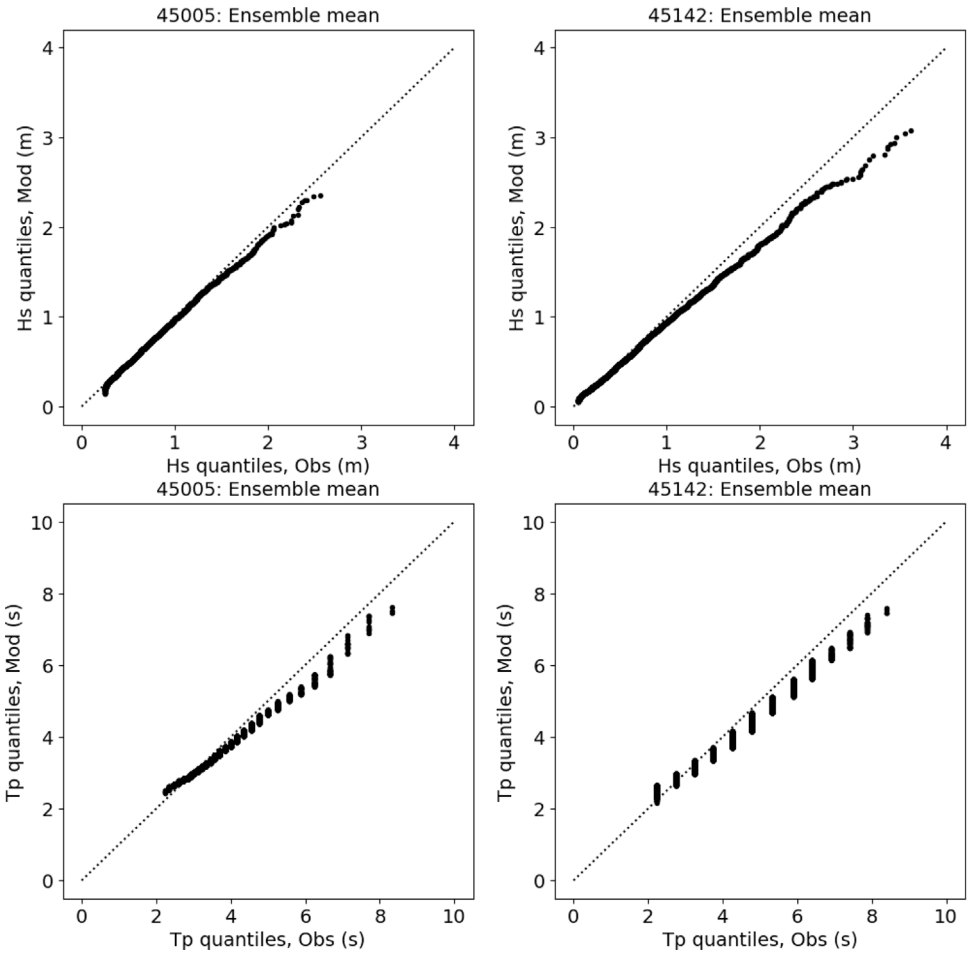


Fig. 23. Q–Q plots of observed and ensemble mean significant wave height (top) and peak wave period (bottom) for the test partition (years 2016–2017). Results for NDBC 45005 (left) and 45142 (right).

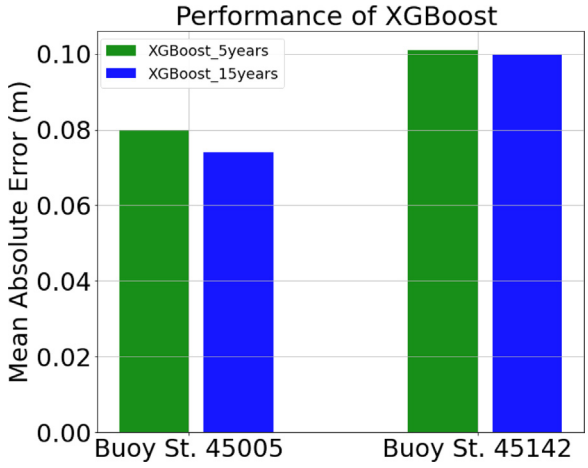


Fig. 24. Mean absolute error in significant wave height of XGBoost for two training periods of different lengths: 5 years (2011–2015) versus 15 years (1994–2015).

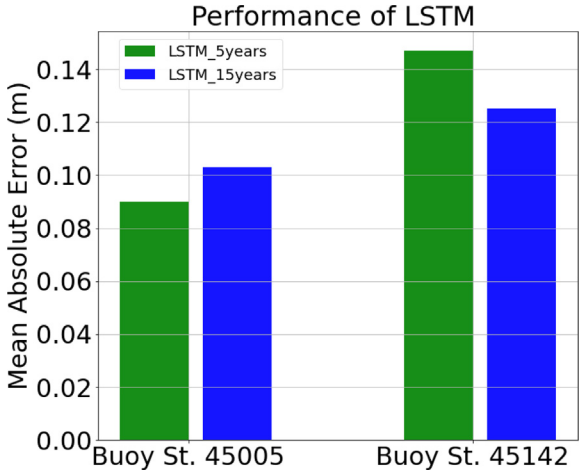


Fig. 25. Mean absolute error in significant wave height of LSTM for two training periods of different lengths: 5 years (2011–2015) versus 15 years (1994–2015).

conditions under which to compare the modeling of wind-wave growth by these models. Buoy data at the upwind NDBC 45005 and downwind NDBC 45142 from 1994 to 2017 were processed for model training, validation and testing. We used 1994–2013 data for training the model, applying observed U_{10} wind velocity as model input, and observed significant wave height and peak wave period as the target variables. Data from 2014–2015 was applied for model validation, and we subsequently used the trained models to evaluate significant wave height and peak period for the unseen test period of 2016–2017. From the results of this study, the following conclusions can be drawn:

1. The physics-based WW3 model was found to perform well in terms of significant wave height, with a low overall bias and

a low scatter, expressed in a Mean Absolute Percentage Error (MAPE) of 15.33% at station 45005 and 21.01% at station 45142. However, WW3 tends to underestimate wave height during storm events, expressed in relative biases of −11.76% and −14.15% at stations 45005 and 45142 respectively, computed over wave heights exceeding the 95th percentile. The scatter in peak wave period was acceptable, with a MAPE of 12.51% and 19.32% at stations 45005 and 45142 respectively. However, the overall relative bias at these stations was quite large, namely −9.02% and −11.51% respectively, increasing to −15.59% and −19.68% during storm peaks.

2. The trained XGBoost model yields significant wave height predictions with comparable scatter to WW3, and extreme values are less underestimated, with a relative bias of −2.56% at station 45005 and −10.61% at 45142 for values above the 95th percentile. Furthermore, the scatter and bias in peak wave period are lower than for WW3, with overall relative bias values of −0.87% and −1.47% at 45005 and 45142 respectively, and values of 9.62% and −10.13% during storm events.
3. The trained LSTM model yields significant wave height predictions with larger scatter and overall bias than WW3 and XGBoost. However, it yields improved relative bias during storm conditions compared to WW3, with values of −3.34% and −8.93% at stations 45005 and 45142 respectively for values exceeding the 95th percentile. For peak wave period, it yields the lowest relative bias of the three models during storm conditions, namely −8.08% and −7.89% for stations 45005 and 45142 respectively. We conclude that for such extreme events, fetch and duration effects are import to incorporate, done here via the LSTM's long- and short-term memories.
4. Ensembling the three studied models was found to be a good strategy, since it combines some of their strengths: The overall scatter of the ensemble mean was lower than any of the component models, with MAPE values of 13.33% and 17.27% in significant wave height, and MAPE values of 7.96% and 12.98% in peak wave period at stations 45005 and 45142 respectively. However, the bias of the ensemble mean fell short of the lowest values amongst the component models: for significant wave height during storm events it yielded relative biases of −5.89% and −11.23% at stations 45005 and 45142 respectively, and for peak wave period during storm events it yielded relative biases −11.10% and −12.57% respectively, which are greater than found for either XGBoost or LSTM.
5. A benefit of the tree-based XGBoost model is that it can also perform well on a smaller set of training data. Training our model on only 5 years of data yields test partition results that are comparable to those found after training the model with the full 15 years of training data. This suggests that the XGBoost model fitting has become saturated after processing about 5 years of training examples.
6. The ML models evaluated here represent a significant reduction in computational times compared to traditional physics-based spectral wave models. To complete the 2016–2017 prediction on the same computing environment, WW3 needed 24 h with 60 CPUs, whereas the trained XGBoost needed only 0.03 s for predicting with 1 CPU, and the trained LSTM required 0.24 s on 1 CPU.

It should be mentioned, however, that the XGBoost and LSTM models built here output only significant wave height and peak period, and only at the point locations for which they were trained. By contrast, WW3 is capable of outputting complete fields of significant wave height, wave period, wave direction, and the full directional wave spectrum. Nonetheless, James et al. (2018) have shown that ML approaches as studied here can readily be extended to outputting fields of wave variables, if sufficient area-covering training data would be available. These authors show how significant wave height results from physics-based models such as SWAN can be used for this purpose. However, biases from these physics-based models, such as those shown in Figs. 18–21, would need to be removed before using their output as training data for the ML models. In this regard, the reader is reminded that the comparisons presented here were made with respect to the operational WW3 model for the Great Lakes as benchmark. It is possible that the performance of the physics-based WW3 could be improved for Lake Erie with a local tuning using the data utilized for training our ML models.

We note furthermore that even though this study was conducted in an enclosed basin to isolate the process of wind-wave growth, the XGBoost and LSTM models evaluated here are equally applicable to open coast and offshore conditions. For such more generalized conditions, currents and offshore wave boundary conditions would have to be included as predictors. For shallow water coastal applications, water level variation from tides and surge should also be included as predictors.

CRediT authorship contribution statement

Haoguo Hu: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft. **André J. van der Westhuysen:** Conception and design of study, Analysis and/or interpretation of data, Writing - original draft. **Philip Chu:** Conception and design of study, Writing - review & editing. **Ayumi Fujisaki-Manome:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft.

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.

Acknowledgments

This study was funded by the NOAA Great Lakes Environmental Research Laboratory awarded to the Cooperative Institute for Great Lakes Research (CIGLR) through the NOAA Cooperative Agreement with the University of Michigan (NA17OAR4320152). This is GLERL contribution 1981 and CIGLR contribution 1182. The authors would also like to thank the reviewers for their constructive comments, which improved the quality of this manuscript.

References

Agrawal, J.D., Deo, M.C., 2002. On-line wave prediction. Mar. Struct. 15, 57–74.

Alves, J.H.G.M., Chawla, A., Tolman, H.L., Schwab, D.J., Lang, G., Mann, G., 2014. The operational implementation of a great lakes wave forecasting system at NOAA/NCEP. Weather Forecast. 29 (6), 1473–1497. http://dx.doi.org/10.1175/WAF-D-12-00049.1.

Ardhuin, F., Rogers, W.E., Babanin, A.V., Filipot, J., Magne, R., Roland, A., Van der Westhuysen, A., Queffeulou, P., Lefevre, J., Aouf, L., Collard, F., 2010. Semiempirical dissipation source functions for ocean waves. Part I: Definition, calibration, and validation. J. Phys. Oceanogr. 40, 1917–1941.

Battjes, J.A., Janssen, J.P.F.M., 1978. Energy loss and set-up due to breaking of random waves. In: Proc. 16th Int. Conf. Coastal Eng.. ASCE, pp. 569–588.

Benjamin, S.G., Brown, J.M., Smirnova, T.G., 2016b. Explicit precipitation-type diagnosis from a model using a mixed-phase bulk cloud–precipitation microphysics parameterization. Weather Forecast. 31, 609–619. http://dx.doi.org/10.1175/WAF-D-15-0136.1.

Benjamin, S.G., Weygandt, S.S., Brown, J.M., Hu, M., Alexander, C.R., Smirnova, T.G., Olson, J.B., James, E.P., Dowell, D.C., Grell, G.A., Lin, H., Peckham, S.E., Lorraine-Smith, T., Moninger, W.R., Kenyon, J.S., Manikin, G.S., 2016a. A North American hourly assimilation and model forecast cycle: The rapid refresh. Mon. Weather Rev. 144, 1669–1694. http://dx.doi.org/10.1175/MWR-D-15-0242.1.

Booij, N., Ris, R., Holthuijsen, L.H., 1999. A third-generation wave model for coastal regions: 1. Model description and validation. J. Geophys. Res.: Oceans 104 (C4), 7649–7666.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Bretschneider, C.L., 1958. Revisions in wave forecasting: deep and shallow water. In: Proc 6th Conf. Coastal Engineering. Council on Wave Research, University of California, pp. 30–67.

Campos, R.M., Krasnopolsky, V., Alves, J.H.G.M., Penny, S.G., 2020. Improving NCEP's global-scale wave ensemble averages using neural networks. Ocean Model. 149, http://dx.doi.org/10.1016/j.ocemod.2020.101617.

Cavaleri, L., 2009. Wave modeling - missing the peaks. J. Phys. Oceanogr. 39 (11), 2757–2778.

Chen, C., Beardsley, R.C., Cowles, G., et al., 2013. An Unstructured Grid, Finite-Volume Community Ocean Model, FVCOM User Manual, SMAST/UMASSD-13-0701, fourth ed. p. 404.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. https://arxiv.org/pdf/1603.02754.pdf.

Chollet, F., 2017. Deep Learning with Python. Manning, Shelter Island, NY, p. 384.

Deo, M., Jha, A., Chaphekar, A., Ravikant, K., 2001. Neural networks for wave forecasting. Ocean Eng. 28 (7), 889–898.

Deo, M., Sridhar Naidu, C., 1999. Real time wave forecasting using neural networks. Ocean Eng. 26 (3), 191–203.

Dixit, P., Londhe, S., 2016. Prediction of extreme wave heights using neuro wavelet technique. Appl. Ocean Res. 58, 241–252. http://dx.doi.org/10.1016/j.apor.2016.04.011.

Etemad-Shahidi, A., Mahjoobi, J., 2009. Comparison between M5' model tree and neural networks for prediction of significant wave height in Lake Superior. Ocean Eng. 36 (15–16), 1175–1181.

Feng, X., Ma, G., Su, S.-F., Huang, C., Boswell, M.K., Xue, P., 2020. A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan. Ocean Eng. 211, 107526. http://dx.doi.org/10.1016/j.oceaneng.2020.107526.

GLERL, 2020. NOAAPORT—Realtime great lakes weather data and marine observations. Retrieved February 1, 2020, from https://coastwatch.glerl.noaa.gov/marobs/.

Hasselmann, K., Barnett, T.P., Bouws, E., Carlson, H., Cartwright, D.E., Enke, K., Ewing, J.A., Gienapp, H., Hasselmann, D.E., Kruseman, P., Meerburg, A., Muller, O., Olbers, D.J., Richter, K., Sell, W., Walden, H., 1973. Measurement of windwave growth and swell decay during the Joint North SeaWave Project (JONSWAP). Dtsch. Hydrogr. Z. Suppl. A (8), 12, 95 pp.

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.

James, S., Zhang, Y., O'Donncha, F., 2018. A machine learning framework to forecast wave conditions. Coast. Eng. 137, 1–10. http://dx.doi.org/10.1016/j.coastaleng.2018.03.004.

Kumar, N.K., Savitha, R., Al Mamun, A., 2018. Ocean wave height prediction using ensemble of extreme learning machine. Neurocomputing 277, 12–20. http://dx.doi.org/10.1016/j.neucom.2017.03.092.

Londhe, S.N., Shah, S., Dixit, P.R., Balakrishnan Nair, T.M., Sirisha, P., Jain, R., 2016. A coupled numerical and artificial neural network model for improving location specific wave forecast. Appl. Ocean Res. 59, 483–491. http://dx.doi.org/10.1016/j.apor.2016.07.004.

Mahjoobi, J., Etemad-Shahidi, A., 2008. An alternative approach for the prediction of significant wave heights based on classification and regression trees. Appl. Ocean Res. 30, 172–177.

Mahjoobi, J., Shahidi, A.I., Kazeminezhad, M.H., 2008. Hindcasting of wave parameters using different soft computing methods. Appl. Ocean Res. 30 (1), 28–36.

Mahmoodi, K., Ghassemi, H., Nowruzi, H., 2017. Data mining models to predict ocean wave energy flux in the absence of wave records. Sci. J. Marit. Univ. Szczec. 49 (121), 119–129. http://dx.doi.org/10.17402/209.

Makarynskyy, O., 2006. Neural pattern recognition and prediction for wind wave data assimilation. Pac. Oceanogr. 3 (2), 76–85.

Malekmohamadi, I., Bazargan-Lari, M.R., Kerachian, R., Nikoo, M.R., Fallahnia, M., 2011. Evaluating the efficacy of SVMs, BNs, ANNs and ANFIS in wave height prediction. Ocean Eng. 38, 487–497.

Mandal, S., Prabaharan, N., 2006. Ocean wave forecasting using recurrent neural networks. Ocean Eng. 33 (10), 1401–1410. http://dx.doi.org/10.1016/j.oceaneng.2005.08.007.

Merembayeva, T., Yunussovb, R., Yedilkhana, A., 2019. Machine learning algorithms for stratigraphy classification on uranium deposits. Procedia Comput. Sci. 150, 46–52.

Moeini, M.H., Etemad-Shahidi, A., 2007. Application of two numerical models for wave hindcasting in Lake Erie. Appl. Ocean Res. 29 (3), 137–145. http://dx.doi.org/10.1016/j.apor.2007.10.001.

NDBC, 2020. National Data Buoy Center. Retrieved February 1, 2020, from https://www.ndbc.noaa.gov/.

Niu, Q., Xia, M., 2016. Wave climatology of Lake Erie based on an unstructured-grid wave model. Ocean Dyn. 66, 1271. http://dx.doi.org/10.1007/s10236-016-0982-7.

Olah, C., 2015. Understanding LSTM networks. Retrieved March 25, 2020, from https://colah.github.io/posts/2015-08-Understanding-LSTMs.

Peres, D.J., Iuppa, C., Cavallaro, L., Cancelliere, A., Foti, E., 2015. Significant wave height record extension by neural networks and reanalysis wind data. Ocean Model. 94, 128–140. http://dx.doi.org/10.1016/j.ocemod.2015.08.002.

Pirhooshyaran, M., Scheinberg, K., Snyder, L.V., 2020. Feature engineering and forecasting via derivative-free optimization and ensemble of sequence-to-sequence networks with applications in renewable energy. Energy 196, 117136. http://dx.doi.org/10.1016/j.energy.2020.117136.

Pirhooshyaran, M., Snyder, L.V., 2020. Forecasting, hindcasting and feature selection of ocean waves via recurrent and sequence-to-sequence networks. Ocean Eng. 207, 107424. http://dx.doi.org/10.1016/j.oceaneng.2020.107424.

Prahlada, R., Deka, P.C., 2015. Forecasting of time series significant wave height using wavelet decomposed neural network. Aquat. Procedia 4, 540–547.

Qi, J., Chen, C., Beardsley, R.C., Perrie, W., Cowles, G.W., Lai, Z., 2009. An unstructured-grid finite-volume surface wave model (FVCOM-SWAVE): Implementation, validations and applications. Ocean Model. 28, 153–166. http://dx.doi.org/10.1016/j.ocemod.2009.01.007.

Qian, Q., Jia, X., Lin, H., 2020. Machine learning models for the seasonal forecast of winter surface air temperature in north america. Earth Space Sci. 7 (8), http://dx.doi.org/10.1029/2020EA001140.

Roland, A., Ardhuin, F., 2014. On the developments of spectral wave models: numerics and parameterizations for the coastal ocean. Ocean Dyn. 64, 833–846.

Schwab, D.J., Bennett, J.R., Liu, P.C., Donelan, M.A., 1984. Application of a simple numerical wave prediction model to lake erie. J. Geophys. Res.: Oceans (1978–2012) 89 (C3), 3586–3592.

Shumway, R.H., Stoffer, D.S., 2017. Time Series Analysis and its Applications: With R Examples, fourth ed. In: Chapter 5.4: Threshold Models, Springer, Cham, Switzerland.

Sørensen, O.R., Kofoed-Hansen, H., Rugbjerg, M., Sørensen, L.S., 2004. A third generation spectral wave model using an unstructured finite volume technique. In: Proceedings of the 29th Int. Conf. on Coastal Engineering. Lisbon, Portugal.

Srinivasan, R., Abhigna, P., Jerritta, S., Rajendran, V., 2017. Analysis of feed forward and recurrent neural networks in predicting the significant wave height at the moored buoys in bay of bengal. In: 2017 International Conference on Communications and Signal Processing (ICCSP). IEEE, pp. 1856–1860.

Tolman, H.L., 2002. Alleviating the garden sprinkler effect in wind wave models. Ocean Model. 4, 269–289.

Tolman, H.L., 2013. A generalized multiple discrete interaction approximation for resonant four-wave nonlinear interactions in wind wave models with arbitrary depth. Ocean Model. 70, 11–24.

Tolman, H.L., Balasubramaniyan, B., Burroughs, L., Chalikov, D., Chao, Y., Chen, H., Gerald, V., 2002. Development and implementation of wind-generated ocean surface wave models at NCEP. Weather Forecast. 17 (2), 311–333.

WAMDI Group, 1988. The WAM model – A third generation ocean wave prediction model. J. Phys. Oceanogr. 18, 1775–1810.