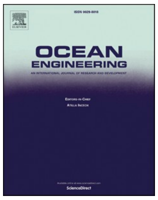
Contents lists available at ScienceDirect

Ocean Engineering

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



A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan

Xi Feng [a,b,*], Gangfeng Ma [c,**], Shih-Feng Su [d], Chenfu Huang [e], Maura K. Boswell [c], Pengfei Xue [e]

[a] *Key Laboratory of Coastal Disaster and Defence (Hohai University), Ministry of Education, China*
[b] *College of Harbor, Coastal and Offshore Engineering, Hohai University, Nanjing, China*
[c] *Department of Civil and Environmental Engineering, Old Dominion University, Norfolk, VA, USA*
[d] *Department of Water Resources and Environmental Engineering, Tamkang University, Taipei, Taiwan*
[e] *Department of Civil and Environmental Engineering, Michigan Technological University, Houghton, MI, USA*

ARTICLE INFO

Keywords:
Lake Michigan
Machine learning
Multi-layer perceptron
Wave forecasting

ABSTRACT

A machine learning framework based on a multi-layer perceptron (MLP) algorithm was established and applied to wave forecasting in Lake Michigan. The MLP model showed desirable performance in forecasting wave characteristics, including significant wave heights and peak wave periods, considering both wind and ice cover on wave generation. The structure of the MLP regressor was optimized by a cross-validated parameter search technique and consisted of two hidden layers with 300 neurons in each hidden layer. The MLP model was trained and validated using the wave simulations from a physics-based SWAN wave model for the period 2005–2014 and tested for wave prediction by using NOAA buoy data from 2015. Sensitivity tests on hyperparameters and regularization techniques were conducted to demonstrate the robustness of the model. The MLP model was computationally efficient and capable of predicting characteristic wave conditions with accuracy comparable to that of the SWAN model. It was demonstrated that this machine learning approach could forecast wave conditions in 1/20,000th to 1/10,000th of the computational time necessary to run the physics-based model. This magnitude of acceleration could enable efficient wave predictions of extremely large scales in time and space.

1. Introduction

Surface waves are an important hydrodynamic component in coastal and ocean engineering designs. A general approach for determining design wave conditions is to estimate high quantiles given certain probabilities of exceedance in the distributions of long-term wave data using extreme value analysis or peak-over-threshold analysis (Soares and Scotto, 2007). However, these analyses often suffer from a scarcity of data because field observations of characteristic wave conditions (e.g., wave height and period) are usually conducted at few buoy stations for a short period of time. As such, wave hindcasting and forecasting become fundamentally important as they provide engineers and scientists with necessary long-term wave information to make planning decisions, design coastal structures, and assess coastal hazards. In addition, long-term wind wave databases are extremely valuable for assessing changing trends of the ocean wave climate (Chawla et al., 2013; Erikson et al., 2015), which can be obtained from wave hindcasting and forecasting using reliable wave models.

Wave hindcasting and forecasting essentially predict characteristic wave heights and periods using readily available weather conditions, i.e., wind speed, storm duration, fetch length, and ice cover. Historically, parametric models, known as SMB methods, are developed for wave forecasting. These models include a set of empirical formulae that relate wave characteristics to wind conditions and water depth in the generating area. They can provide efficient, but rough estimates of characteristic wave heights as well as wave periods for structural design purposes (Goda, 2010). The current practice of wave forecasting usually employs a numerical model for the computation of directional wave spectrum. The most popular ocean wave models are the third-generation spectral wave model WaveWatch III (Tolman et al., 2002) and the nearshore wave spectral model SWAN (Booij et al., 1999). WaveWatch III incorporates formulations for the deep-water wave processes, such as wave generation by wind, energy dissipation by whitecapping, and the quadruplet wave-wave interactions, and is applicable for deep-ocean

* Corresponding author.

** Corresponding author. Key Laboratory of Coastal Disaster and Defence (Hohai University), Ministry of Education, 1 Xikang Road, Nanjing, 210098, China.
E-mail addresses: xifeng@hhu.edu.cn (X. Feng), gma@odu.edu (G. Ma).

https://doi.org/10.1016/j.oceaneng.2020.107526
Received 11 April 2019; Received in revised form 21 April 2020; Accepted 11 May 2020

wave simulations at a global scale. On the other hand, the SWAN wave model is focused on wave transformation and deformation in shallow water. It incorporates shallow-water wave processes, such as wave energy dissipation due to bottom friction and breaking, triad wave-wave interactions, and interactions between waves and ambient currents. These models were used for wave hindcasting and forecasting in studies including Mori et al. (2010), Chawla et al. (2013), Erikson et al. (2015), Kukulka et al. (2017), Niroomandi et al. (2018), and Allahdadi et al. (2019), to list a few. Although wave spectral models are capable of accurately simulating wave fields, they are computationally expensive.

The machine learning (ML) approach was found capable of efficiently mapping large datasets to quantities of interest and have been widely used for forecasting in geosciences (Lary et al., 2016; DeVries et al., 2017), hydrology (Hong, 2008; Ahmad et al., 2010; Rasouli et al., 2012), and engineering (Lee, 2006; Etemad-Shahidi and Bonakdar, 2009; Etemad-Shahidi et al., 2011). For wave forecasting, due to a lack of large datasets for training, ML is used to forecast characteristic wave conditions at a few specific locations (Deo et al., 2001; Tsai et al., 2002; Gunaydin, 2008; Malekmohamadi et al., 2011). Recently, James et al. (2018) developed an ML framework for wave forecasting at a large domain in Monterey Bay. The ML algorithms were trained on a large dataset produced by the physics-based wave model SWAN. They found that the ML models yielded wave heights and wave periods consistent with the SWAN model. In addition, the ML models could dramatically accelerate wave simulations by more than 4000 times. O'Donncha et al. (2018, 2019) presented a framework that integrated physics-based models with an ML algorithm and combines forecasts from multiple, independent models into a single "best-estimate" prediction of wave conditions. They showed that the framework, which integrated data-driven and physics-based approaches, could outperform either technique in isolation. These studies demonstrated that ML is a promising tool for improving wave forecasting at large spatial and temporal scales.

This paper describes an ML framework based on a multi-layer perceptron (MLP) learning algorithm to hindcast and forecast characteristic wave conditions in Lake Michigan, where wave dynamics are significantly affected by ice coverage during the winter season. The MLP algorithm was selected for both wave height and period forecasting, which is a nonlinear regression problem that relates wave characteristics to weather conditions. The MLP algorithm was trained and validated using data hindcasted from a physics-based wave model (SWAN) for the period 2005–2014. Wave forecasting was performed for the year 2015 to examine the model performance. This study shows that the MLP approach can radically accelerate wave hindcasting and forecasting, while retaining predictive accuracy comparable to physics-based wave modeling. The paper is organized as follows. Section 2 presents the physics-based wave modeling and the quality of SWAN model results. The MLP algorithm is introduced in section 3. Training and validation of the algorithm, as well as its performance on wave forecasting in Lake Michigan, are also presented in this section. Section 4 discusses the performance of the MLP model on wave prediction considering multiple factors, e.g. length of training dataset, hyperparameters, and etc. The paper is concluded in section 5.

2. Physics-based wave modeling

2.1. SWAN model

To obtain training and validation datasets for supervised MLP, wave simulations for Lake Michigan using the physics-based nearshore wave model SWAN v41.20 were completed. SWAN is a third-generation spectral wave model, developed at Delft University of Technology, that computes random/irregular, shore-crested wind-generated waves in coastal regions and inland waters (Booij et al., 1999). It solves the evolution equation of action density $N(\vec{x}, t, \sigma, \theta)$ in space $\vec{x}$ and time t as well as wave-energy distribution over frequencies σ and propagation directions θ. The action density is defined as $N = E/\sigma$, where E is the wave-energy density. The evolution of wave-action density is governed by (Komen et al., 1994):

$$\frac{\partial N}{\partial t} + \nabla_{\vec{x}} \cdot \left[\left(\vec{c}_g + \vec{U} \right) N \right] + \frac{\partial_{c_\sigma} N}{\partial \sigma} + \frac{\partial_{c_\theta} N}{\partial \theta} = \frac{S_{\text{tot}}}{\sigma}, \tag{1}$$

where $\vec{U}$ is the current velocity vector, $\vec{c}_g = \frac{\partial \theta}{\partial \vec{k}}$ is the group velocity, $\vec{k}$ is the wave-number vector, c_σ and c_θ are the propagation velocities in spectral space (σ, θ), respectively. S_{tot} accounts for wave energy sources and sinks, including wave generation by wind, wave decay due to whitecapping, bottom friction, and depth-induced wave breaking, and energy redistribution through nonlinear wave-wave interactions.

2.2. Wind data

In the SWAN wave model, high-quality wind inputs are critically important for wave simulations. In this study, the wind data Climate Forecast System Reanalysis (CFSR) (Saha et al., 2010, 2014) from the National Centers for Environmental Prediction were employed. The CFSR uses a coupled atmosphere-ocean-land surface-sea ice system with advanced data assimilation techniques and an extensive database of meteorological observations to create its products. The original CFSR dataset spans from 1979 to 2010 and the second version of the Climate Forecast System (CFSRv2) provides products from 2011 to present with several improvements over CFSR, such as a higher spatial resolution (Saha et al., 2014). The temporal resolution of CFSR wind is 6 h with-more detailed information available in Saha et al. (2010, 2014). At each time step, the SWAN modeling system automatically interpolated the CFSR dataset in time and space and interpolates the CFSR wind field of one time snapshot spatially on the SWAN grid domain for resolving Equ. (1). The CFSR wind data were also taken into the MLP for data training, validation, and prediction. In the MLP, the CFSR wind data were interpolated spatially onto a coarser grid (78 grid resolution), and are temporally assembled into one matrix. The detailed MLP algorithm is described in section 3.1. Fig. 1 is the schematic plot that illustrates the technical route of the ML framework. In Fig. 1, a SWAN model system of Lake Michigan was set up at first. The SWAN model was driven by the CFSR reanalysis wind fields and incorporated ice coverage information. These 'features', along with SWAN outputs of spatial variables (wave characteristics), were aggregated into training and validating datasets, which were supplied to machine learning models.

2.3. Ice data

Wind-driven waves in Lake Michigan can be significantly impacted by ice. To include the effect of ice, historical ice fields were incorporated into wave simulations by employing a technique that eliminated wave dynamics when ice coverage was higher than a threshold value. Previous studies implemented threshold values varying from 30 to 50% (Hubertz et al., 1991; Bennington et al., 2010; Anderson et al., 2015). In this study a threshold of 30% was selected, as suggested by Anderson et al. (2015), meaning that wave dynamics were not considered and the water depth is manually set to zero when ice coverage exceeded 30%. Historical ice data for the lake are accessible via NOAA Great Lakes Ice Atlas and Great Lakes Environmental Research Laboratory. The data have a horizontal resolution of ~2.5 km prior to 2007 and are upgraded to a resolution of ~1.8 km in 2007. More details regarding the ice datasets are available (Assel et al., 2003, 2013; Assel, 2005).

2.4. Model configuration and validation

Wave simulations in Lake Michigan were performed from 1979 to 2015; the 37 years when wind and ice data are available. Fig. 2 shows the computational domain of the SWAN model, along with the locations

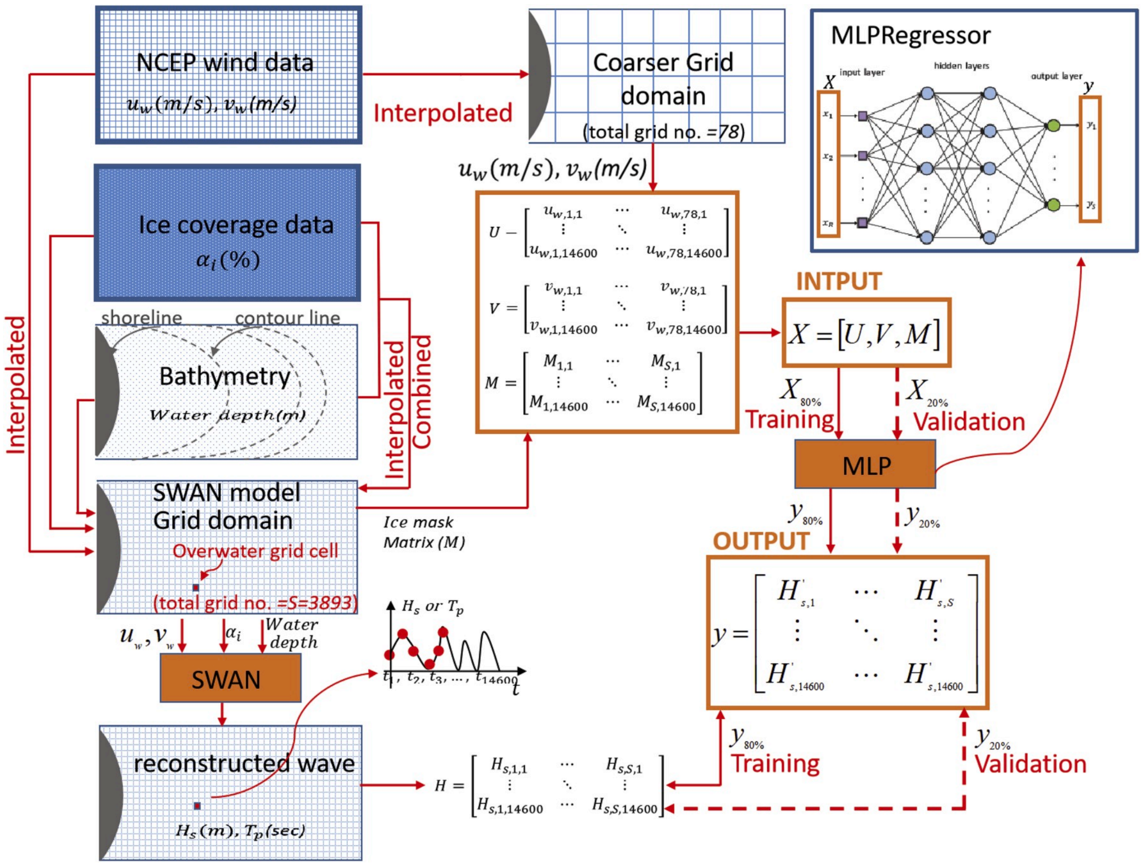

NCEP wind data
$u_w(m/s), v_w(m/s)$
Interpolated
Coarser Grid domain
(total grid no. =78)
$u_w(m/s), v_w(m/s)$
MLPRegressor
X
input layer
hidden layers
output layer
y
x_1
x_2
x_R
y_1
y_S
Ice coverage data
$\alpha_i(\%)$
shoreline
contour line
Bathymetry
Water depth(m)
Interpolated
Interpolated
Combined
$U = \begin{bmatrix} u_{w,1,1} & \cdots & u_{w,78,1} \\ \vdots & \ddots & \vdots \\ u_{w,1,14600} & \cdots & u_{w,78,14600} \end{bmatrix}$
$V = \begin{bmatrix} v_{w,1,1} & \cdots & v_{w,78,1} \\ \vdots & \ddots & \vdots \\ v_{w,1,14600} & \cdots & v_{w,78,14600} \end{bmatrix}$
$M = \begin{bmatrix} M_{1,1} & \cdots & M_{S,1} \\ \vdots & \ddots & \vdots \\ M_{1,14600} & \cdots & M_{S,14600} \end{bmatrix}$
INTPUT
$X = [U, V, M]$
$X_{80\%}$
Training
$X_{20\%}$
Validation
MLP
SWAN model Grid domain
Overwater grid cell
(total grid no. =S=3893)
Ice mask Matrix (M)
$y_{80\%}$
$y_{20\%}$
OUTPUT
$y = \begin{bmatrix} H'_{s,1} & \cdots & H'_{s,S} \\ \vdots & \ddots & \vdots \\ H'_{s,14600} & \cdots & H'_{s,14600} \end{bmatrix}$
H_s or T_p
$t_1, t_2, t_3, \ldots, t_{14600}$
t
u_w, v_w
α_i
Water depth
SWAN
reconstructed wave
$H_s(m), T_p(sec)$
$H = \begin{bmatrix} H_{s,1,1} & \cdots & H_{s,S,1} \\ \vdots & \ddots & \vdots \\ H_{s,1,14600} & \cdots & H_{s,S,14600} \end{bmatrix}$
$y_{80\%}$
Training
$y_{20\%}$
Validation

Fig. 1. Schematic plot of training and validation processes for wave forecasting using the multi-layer perceptron regressor (MLP) approach, in which reconstructed wave conditions in Lake Michigan modeled from SWAN served as the input ('features' in ML terms).

of two NOAA buoy stations. Measurements of significant wave height (H_s) and peak wave period (T_p) at these two buoys were used to validate the SWAN model and the MLP framework. The bathymetric data in Fig. 2 were obtained from the NOAA National Geophysical Data Center and interpolated on the computational mesh grid. The computational mesh in Fig. 2 is curvilinear and comprised of 180 × 40 grid cells and 181 × 41 grid points, including both overwater and overland grid points. In all simulations, wave energy dissipation mechanisms including whitecapping, depth-induced wave breaking, and bottom friction, were activated. The spectral domain was discretized into 12 directions and 31 frequency bands.

To develop reliable training data for the MLP model, it is important to demonstrate that SWAN can accurately replicate wave conditions in Lake Michigan. Long-term wave measurements at two NOAA buoy

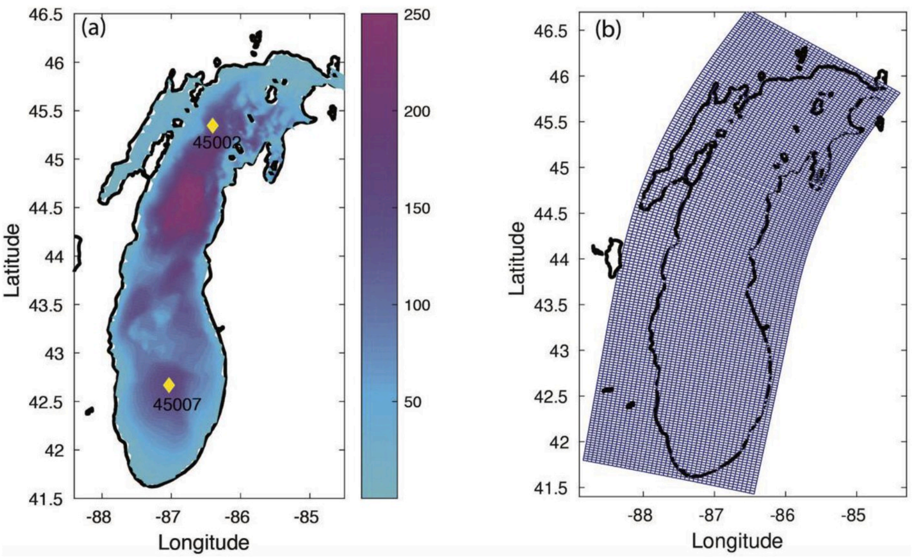


Fig. 2. (a) Lake Michigan and the locations of two NOAA buoy stations, contour color represents bathymetry (in unit of m); (b) The computational mesh for the SWAN model. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

stations, 45002 and 45007, were used for SWAN model validation, spanning the entire 37 years of available data. For demonstration, the simulated and measured H_s and T_p at buoy stations 45002 and 45007 were compared in Fig. 3. To quantify model performance, the bias ($= \frac{1}{N}\sum_{i=1}^{N}(M_i - S_i)$, where M_i is the measurement and S_i is the simulation), the root-mean-square-error (RMSE), and the coefficient of determination R^2 at both buoy stations were evaluated for 2014. As demonstrated in Fig. 3, the model predicted the H_s quite well with R^2 scores above 0.6 and small biases and RMSEs, which are comparable to similar studies (e.g., Niroomandi et al., 2018; James et al., 2018; Allahdadi et al., 2019; Shi et al., 2019; Kutupoğlu et al., 2018; Kukulka et al., 2017; Akpinar et al., 2016). The model performance on T_p is relatively weak with low R^2 scores. Considering small temporal variations in T_p, these scores are reasonable since the correlation coefficient measures the predicted proportion of the variance in the data. Additionally, the biases and RMSEs of the predicted T_p are acceptable (e.g. a bias of-0.006s for 45002 and 0.267s for 45007), when compared to a SWAN simulation, and comparable to similar studies.

From the aforementioned evaluations it could be argued that the SWAN model is capable of reasonably simulating temporal variations of H_s and T_p. In addition, the SWAN model has a lower utilization rate of computational resources when compared to a wave-resolving model. Discrepancies between the measurements and modelling results could arise from a variety of systematic errors, including measurement instruments, modelling systems, and wind data. For instance, some large wave events were underestimated by the SWAN model. This underestimation could possibly be caused by the coarse 6-h temporal resolution of the CFSR wind dataset (Saha et al., 2010, 2014). It is likely that certain large wind events are not captured by the data (e.g. Li et al., 2016; Shi et al., 2019; Allahdadi et al., 2019). As a result, the wave heights during these events were underestimated. Additionally, the spatial resolution (0.3° for 1979–2010 and 0.2° for 2011-present day) would also cause underestimation of local wind-waves, particularly for locations close to the land boundary. Previous studies also reported the mismatches in H_s and T_p, even after calibration of the physics-based wave models (e.g. Allahdadi et al., 2019; Shi et al., 2019; Kutupoğlu et al., 2018; Kukulka et al., 2017; Akpinar et al., 2016). Compared to other datasets, such as the ERA-Intrim reanalysis data from ECMWF, CFSR shows better performance in wave modeling at North Atlantic and European shelves (e.g. Allahdadi et al., 2019; Kukulka et al., 2017; James et al., 2018).

The distributions of SWAN-simulated yearly-averaged H_s and T_p are shown in Figs. 7a and 8a, which provide global portraits of the wave field over the lake. The yearly-averaged wave height is generally larger in the northern lake than that in the southern lake and higher in the deep water. The yearly-averaged T_p has low variations with around 4.0 s in most deep-water regions. The wave period is also longer in the northern lake and shorter in shallow water.

2.5. Machine learning for wave forecasting

The ML approach involves a set of statistical tools and algorithms for data modeling. Algorithms include linear and logistic regression, decision trees and support vector machines boosting, multi-layer perceptrons, etc. Depending on whether there are targeted outputs, ML problems generally fall into two broad categories: supervised learning, in which the model is presented with inputs and targeted outputs, where it learns to map inputs (often called ‘features’ in ML terms) to outputs (often called ‘labels’ in ML terms); and unsupervised learning, where no outputs are given to the learning algorithm, and the goal is to find the structure in the inputs. Supervised ML applications can be further classified into regression problems, which have a quantitative response or output, and classification problems, which typically have a qualitative or categorical response.

It is obvious that wave forecasting is a supervised-regression problem, which maps inputs, such as wind field and ice coverage, to output bulk wave parameters, such as H_s and T_p. In this section, an ML framework is developed to act as an efficient surrogate for the physics-based SWAN wave model (Fig. 1). The framework is based on the popular Python library Scikit-Learn version 0.19.2. Several regression algorithms are available in Scikit-Learn, including linear regression, tree-based regressors (i.e., decision tree regressor, random forest regressor), and an MLP regressor. Wave forecasting is a nonlinear problem, thus linear regression is not appropriate for this application. In tree-based models, the predictor space is first divided into a number of distinct and non-overlapping regions. For every observation that falls into a certain region a corresponding prediction is made, which is simply the mean of the response values for the observation. Therefore, the MLP regressor is selected for the current application.

2.6. Multi-layer perceptron

MLP is a supervised learning algorithm that learns a nonlinear function and maps inputs to outputs by training on a dataset. Given a set of inputs $X = x_1, x_2, \cdots, x_R$, and outputs $y = y_1, y_2, \cdots, y_S$, where R is the number of inputs and S is the number of outputs, the MLP learns a nonlinear function approximator $f(\cdot): X \rightarrow y$ for either classification or regression. The MLP consists of three or more layers (an input layer, an output layer, and one or more hidden layers). Each node in one layer connects with a certain weight to every node in the following layer. The

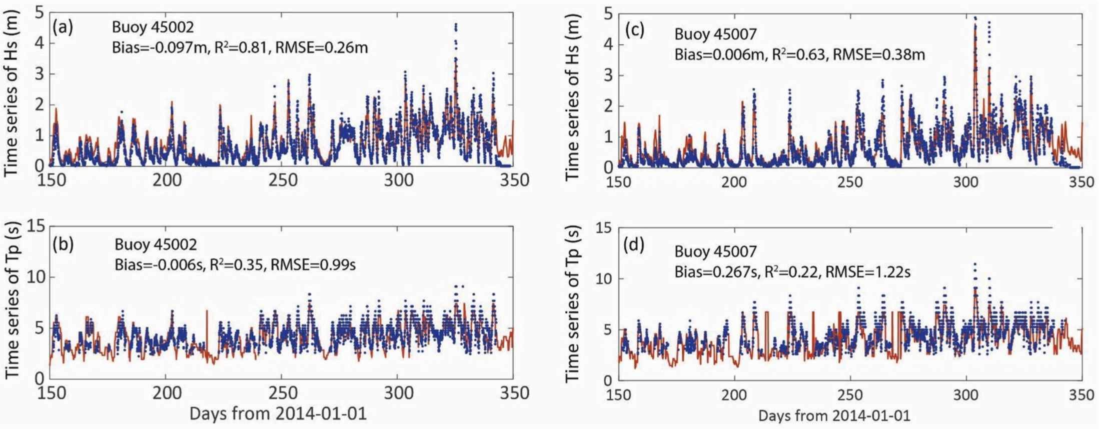


Fig. 3. Comparison of measured and simulated (a,c) significant wave height, $\boldsymbol{H_s}$ and (b,d) peak wave period, $\boldsymbol{T_p}$ at (Left) buoy 45002 and (Right) buoy 45007 in year 2014. Red lines represent SWAN modeled results and blue dots indicate measurements. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

inserted box with blue edgecolor on the upper right corner of Fig. 1 shows the structure of an MLP regressor. The input layer consists of a set of neurons, with X representing the inputs. The rightmost layer is the output layer that receives information from the last hidden layer and transforms it into output values. Each neuron in the hidden layer accumulates the values from the previous layer as a weighted linear summation with a bias, followed by a nonlinear activation function. For example, the output at the j_{th} node of the first hidden layer is given by

$$\mathrm{out} = g\left(\sum_{i=1}^{R} w_{ji}x_i + b_j\right), \tag{2}$$

where $g(\cdot)$ is the nonlinear activation function, w_{ji} is the weight, and b_j is the bias. There are various activity functions in the MLP regressor, including 'identity', 'sigmoid', 'tanh', and 'relu'. This study selected the rectified linear unit (ReLU, or so called 'relu') (Nair and Hinton, 2010) as the activation function, which ensured no negative outputs. As tested during calibration, 'relu' outperformed the other activity functions due to less training- and validating-time; although, the accurate score was not significantly improved.

$$g(\mathscr{Z}) = max(0, \mathscr{Z}). \tag{3}$$

The MLP is trained by adjusting connection weights and biases based on the amount of error in the output compared to the expected result encapsulated in the loss function. This learning process is carried out through forward- and back-propagation and solved by the "adam" optimizer, which is an algorithm for optimization of stochastic objective functions, proposed by Kingma and Ba (2014). The method is straightforward to implement and computationally efficient with little memory requirements. Moreover, it is also invariant to diagonal rescaling of the gradients and well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients, which makes it suitable for wave forecasting.

2.7. Data preprocessing

Input datasets ('features') of the MLP included wind field, water depth, and ice coverage of the lake area. The sources of wind from CFSR, bathymetric, and ice coverage data were presented in section 2. To improve training- and validation-efficiency, the size of X for the MLP regressor was reduced. In the MLP regressor, X was assembled by 10-year long wind and ice coverage data from 2005 to 2014. The reconstructed wave condition (H_s and T_p) from SWAN for 2005–2014 were used for training and validating the MLP regressor outputs y (Fig. 1). $X = [U, V, M]$, where U, V stand for the sub-matrices of wind component in the Cartesian coordinates, and M stands for the sub-matrix of ice mask, which combines ice coverage and bathymetry information. X had 14,600 rows. The number of rows stand the number of snapshots for the 10 years, which is exactly the number the SWAN timesnap outputs (4 times/day x 365 days/year x 10 years). It was determined that a 10-year dataset is sufficient to create an efficient and an accurate MLP regressor for wave forecasting (refer to section 4).

To further reduce the size of the input matrix, the wind data (u_w and v_w) are interpolated into a coarser grid, resulting in 78 wet grid points in Lake Michigan. Thus, the wind vectors are assembled into two sub-matrices: U and V; each one has a size of 14,600 x 78. This treatment could still obtain a good representation of the wind field in Lake Michigan because the spatial variation of wind was insignificant.

The ice coverage sub-matrix M, which determines whether the wave dynamics are taken into account was generated as follows. First, bathymetric and ice coverage data were interpolated into the SWAN grid as shown in Fig. 2. In the MLP, the overland grid points are not considered, which could dramatically reduce the input dataset size. As a result, a total of 3893 overwater grid points in Lake Michigan are considered. In Fig. 1, naming S as the total number of overwater grid points, $S = 7421$. Second, an ice-mask matrix was created. The size of the resulting ice-mask matrix (M) is $14{,}600 \times 3{,}893$. Given an overwater grid point, the mask equals 1. If the ice coverage on the grid point is above 30%, the mask is set to 0. Thus, X had 4049 columns (78 for U +78 for V +3893 for M).

In practice, the input dataset must be preprocessed. Specifically, the wind speeds u_w and v_w were normalized and scaled using their means and standard deviations. The resulting wind inputs have zero means and unit deviations. No preprocessing of the ice-mask matrix and target-vector y is required. The preprocessed X and y were randomly shuffled into two subsets: training and validation. Selection of a portion of the matrices of X and y for validation was based on model performance. The portion was determined to be 0.2 for the model's best performance (refer to section 4). Thus, the training dataset ($X_{80\%}$) is composed of 11,680 rows of X, and the validation dataset ($X_{20\%}$) is composed of the remaining 2920 rows.

In this study, the wave height and wave period models were constructed and trained separately. The wave height model outputs significant wave height (H_s) and the wave period model outputs peak wave period (T_p), following the SWAN outputs. The two bulk parameters are independent of each other physically and by setting up two separate models ensured both H_s and T_p could obtain better predictions. For both models, the target-vector y has dimensions of 14,600 rows (SWAN model runs) and 3893 columns (the number of wet grid points), where y is composed of either H_s or T_p. Accordingly, y is divided into two sub-matrices, $y_{80\%}$ and $y_{20\%}$. The training dataset $y_{80\%}$ has 11,680 rows, corresponding to $X_{80\%}$. The MLP algorithm was trained using the training dataset and then applied to the validation dataset to evaluate its performance.

2.8. Training and validation

2.8.1. Significant wave height

The constructed MLP regressor for H_s was trained using the training dataset. As proposed in section 3.1, the stochastic gradient-based optimizer 'adam', proposed by Kingma and Ba (2014), was used after calibration. In the following results, the parameters of the MLP regressor were optimized by a cross-validated grid-search over a parameter grid, which is implemented by the GridSearchCV function in Scikit-Learn. These parameters, evaluated by GridSearchCV, and their optimized values are concluded in Table 1. The number of hidden layers, one, two, or three, and the number of neurons in each layer, ranging from 100 to 500 in increments of 50, are interrogated. Results were evaluated by the R^2 score and showed that two layers with 300 neurons per layer produced the best fitting results. The sensitivity tests on the important parameters of the MLP regressor are shown in section 4, such as early-stopping, learning rate, size of minibatch, stepsize (α), exponential decay rates for the first and the second momentum estimates (β_1, β_2), etc.

After training, the model performance was further evaluated using the validation dataset. In this procedure, the targeted wave height data in the validation dataset were not used. Instead, they were only used to calculate the error of the predicted H_s. The $X_{20\%}$ were input into the trained MLP regressor to predict the wave heights for the entire Lake Michigan. Fig. 4a shows the cross-plot of spatially-averaged H_s from the SWAN wave model and the MLP model on the validation dataset over the entire lake. Clearly, the MLP model could reasonably replicate SWAN simulations. The RMSE, bias, and R^2 score are 0.10 m, −45.e−04 m, and 0.94, respectively. Fig. 4a indicates that the MLP model slightly overestimates wave heights under calm conditions when H_s is small. Yet, the current MLP model overall is not over fitted as dots are scattered in line with $y = x$.

2.8.2. Peak wave period

The reconstruction of the T_p MLP model is similar to that of the H_s

Table 1
Scikit-Learn MLP regressor parameters optimized and used in this study.

Parameters[a]	Values tested in GridSearchCV	Optimized Value
solver	'adam'	'adam'
hidden_layer_sizes	(100,100),...(350,350),(400,400)	(300,300)
activation	'identity','relu','tanh', 'sigmoid'	'relu'
alpha	0.1,0.01,0.001,0.0001,0.00001	0.0001
beta_1	0,0.5,0.9	0.9
beta_2	0.99,0.999,0.9999	0.999
batch_size	100,200,300,400	200
early_stopping	TRUE, FALSE	FALSE
validation_fraction	0.1,0.2,0.3,0.4,0.5	0.2
learning_rate	'constant', 'invscaling', 'adaptive'	'adaptive'
max_iter	100,200,300,400	100
learning_rate_ini	0.01,0.001	0.001
epsilon = 1e-08	0.00000001	0.00000001
shuffle	TRUE	TRUE
random_state	None	None
tol	0.0001	0.0001
verbose	FALSE	FALSE
warm_start	FALSE	FALSE
n_iter_no_change	10	10

[a] Note the meaning of each parameter listed in the table is as follows: olver: the solver for weight optimization,'adam':the option representing a stochastic gradient-based optimizer proposed by Kingma and Ba (2014), hidden_layer_sizes: the number of neurons in the ith hidden layer, activation: activation function for the hidden layer, alpha: L2 penalty (regularization term) parameter, beta_1: exponential decay rate for estimates of the first moment vector in adam, should be in [0, 1], beta_2: exponential decay rate for estimates of the second moment vector in adam, should be in [0, 1), betch_size: size of minibatches for stochastic optimizers, arly_stopping: terminate the training when validation score is not improving, validation_fraction: the proportion of training data to set aside as the validation set, learning_rate: the learning rate schedule for weight updates, max_iter: maximum number of iterations, learning_rate_ini: the initial learning rate, epsilon: the value for numerical stability in adam, shuffle: whether to shuffle samples in each iteration, random_state: if none, the random number generator is the RandomState instance used by np.random, tol: tolerance for the optimization. When the loss or score is not improving by at least tol for n_iter_no_change consecutive iterations, unless learning_rate is set to 'adaptive', convergence is considered to be reached and training stops.,verbose: whether to print progress messages to stdout, warm_start: when set to True, reuse the solution of the previous call to fit as initialization; otherwise, just erase the previous solution, n_iter_no_change: maximum number of epochs to not meet tol improvement. For the ones not listed, they are either not important and are taken as default by the model or not used when using 'adam' for solver.

MLP model. Using the GridSearchCV, it is determined that two layers with 300 neurons per layer still produced the best validation accuracy. The cross-plot of spatially-averaged T_p from the SWAN wave model and the MLP model for the validation dataset is presented in Fig. 4b. Compared to the SWAN results, the MLP model produces slightly higher estimations for small waves (<1 s) and lower estimations for large waves (>6 s). Further zoom-in analysis on a single node shows that SWAN overestimates T_p for long-period waves, while MLP produces results closer to the observations. On average, predictions of T_p from the MLP model are consistent with those from the SWAN model, as the bias between the two is nearly zero. The RMSE, bias, and R^2 score for spatially-averaged T_p are 0.48 s, −0.004 s, and 0.75, respectively.

2.9. Wave forecasting

Although the MLP models have been tested against characteristic wave conditions in the validation dataset (Section 3.3), it is important to examine the performance of the model on wave forecasting. In this section, the constructed MLP models are applied to forecast wave conditions for 2015. The wind and ice coverage data were preprocessed in the same way as detailed in section 3.2. The measurements at two NOAA buoy stations and the SWAN hindcasted wave conditions were used to examine the performance of the MLP models. Fig. 5 shows the comparisons of H_s and T_p from the MLP model and the SWAN model to the measurements for year 2015. The measurement data are missing beyond around day 325 for buoy 45002 and day 335 for buoy 45007. The MLP model reasonably predicts the temporal variations of H_s observed at both 45002 and 45007. For example, the biases between MLP and observations of H_s are 0.11 m at 45002 and 0.13 m at 45007 and the biases of T_p are 0.17 s at 45002 and -0.18 s at 45007, respectively. In addition, the performance of the MLP model for wave forecasting is comparable to that of the SWAN model at both buoys (Fig. 5).

The spatially averaged MLP model predictions are also compared with wave forecasting by the SWAN model for the entire lake. The correlations between the spatially-averaged MLP model and the SWAN predictions are presented in Fig. 6, which indicates that the forecasted wave conditions by the MLP and SWAN models are closely correlated and fairly similar. The bias and R^2 scores between two predictions were −0.014 m and 0.92 for H_s and −0.010 s and 0.79 for T_p, respectively.

The comparisons of yearly-averaged H_s and T_p from the SWAN wave model and the MLP model are demonstrated in Figs. 7 and 8. The predicted yearly-averaged H_s and T_p from the SWAN and MLP models have similar spatial distributions, except that the SWAN predictions appear more evenly distributed in space. The MLP predictions on wave height in the northern and eastern lake are slightly smaller than the SWAN simulations, while wave height predictions by the MLP model are larger in the southern lake. The differences between these two model predictions

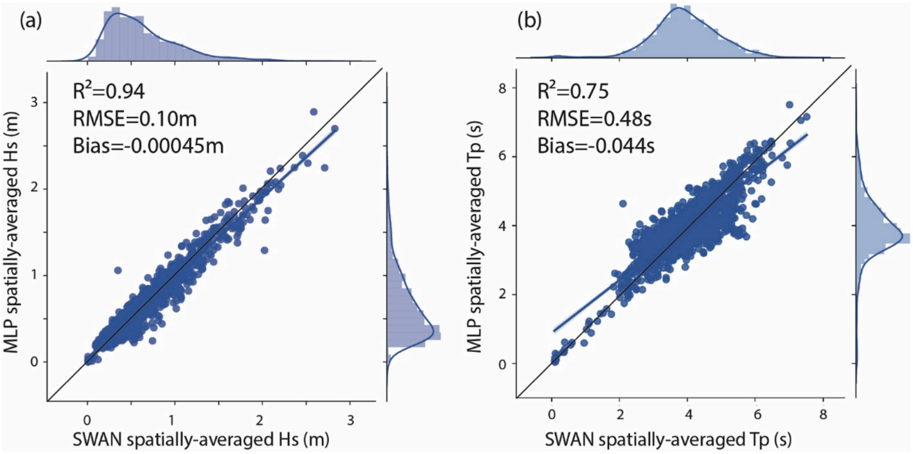


Fig. 4. Cross plots of spatially-averaged (a) significant wave height $\boldsymbol{H_s}$ and (b) peak wave period $\boldsymbol{T_p}$ over Lake Michigan, modeled by SWAN and output from the MLP model during validation process. The thin black line indicates the 1:1 line and the blue line indicates the best-fitting line. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

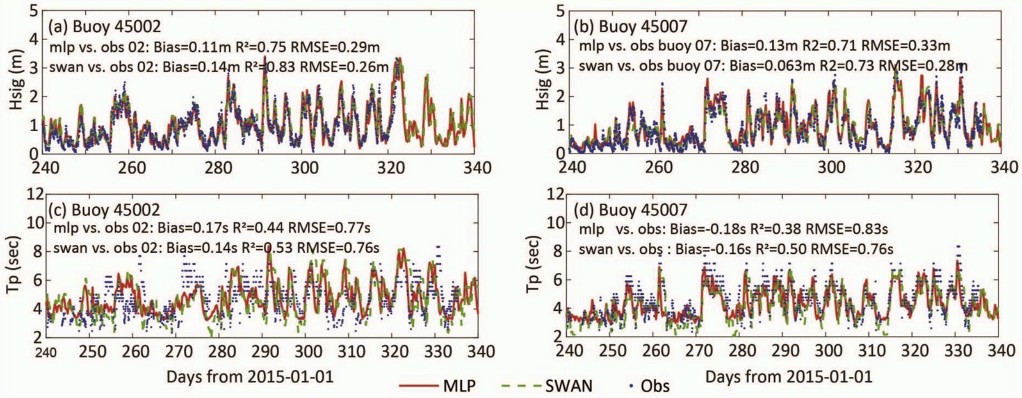


Fig. 5. Comparisons of (a,b) significant wave height $\boldsymbol{H_s}$ and (c,d) peak wave period $\boldsymbol{T_p}$ at two NOAA buoy stations from MLP and SWAN modeled wave forecasting in year 2015 to observations. Red lines represent MLP predictions, green dashed lines indicate SWAN predications, and blue dots indicate observations. (Left panel: buoy 45002 and Right panel: buoy 45007). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

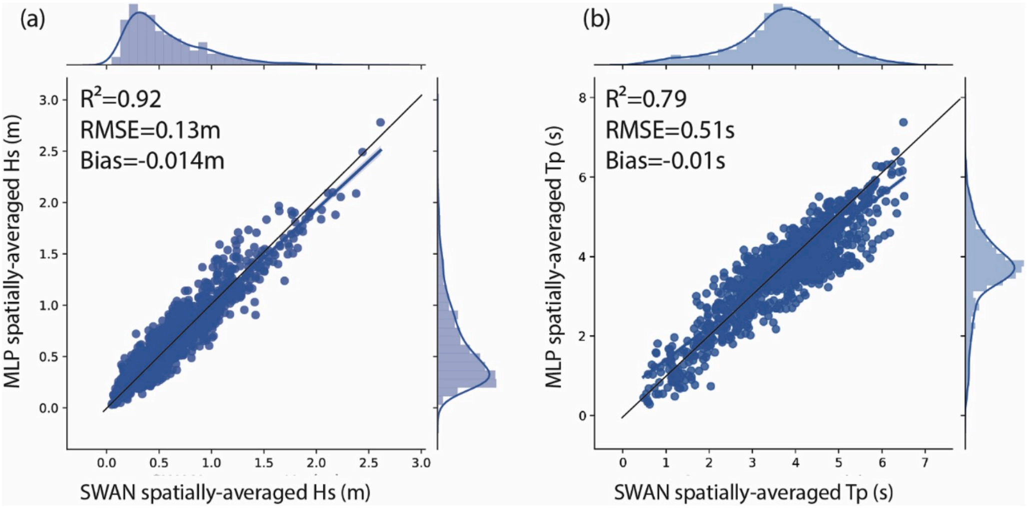


Fig. 6. Cross plots of spatially-averaged (a) significant wave height $\boldsymbol{H_s}$ and (b) peak wave period $\boldsymbol{T_p}$ over Lake Michigan, predicted by the SWAN wave model and the MLP model in year 2015. The thin black line indicates the 1:1 line and the blue line indicates the best-fitting line. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

are generally less than 0.1 m. In terms of T_p, the SWAN predictions are slightly longer in the northern lake near the islands. In most deep water regions, the differences of the yearly-averaged T_p between the two models are less than 0.5 s. Generally, the MLP model has better predictions in open deep water and poor predictions in shallow water. This is because the MLP model could not capture the wave processes around islands and shoreline in the shallow water, such as wave shoaling, refraction, and diffraction.

3. Discussions

The MLP approach was proved to be a promising tool for wave forecasting in Lake Michigan as explained in the previous section. In this section, we present the sensitivity tests on the factors and hyperparameters that are important in the MLP regressor, including the length of the input dataset X ('features'), the fraction of X necessary for training/validation, the early-stopping option, the mini-batch size, the learning rate, and other hyperparameters (α, β_1, β_2).

One challenge to applying such ML approaches to other water bodies is their enormous appetite for data (James et al., 2018). Thus, the effect of data length for training and validation on the MLP model was first assessed. Fig. 9 shows the accuracy scores versus data length for the total training and validation data in years. The hindcasted dataset from the SWAN model for the period 2005–2014 was used here and the training and validation dataset are randomly picked from the entire 10-year long dataset. For example, in Fig. 9, the 1 year-long dataset stands for 1/10 of the 10-year long dataset. In these tests, the MLP model had the same structure with two hidden layers and 300 neurons in each hidden layer. The accuracy score was evaluated by R^2 calculated on the validation dataset, which accounts for 20% of the total training and validation data. It was shown that if the training and validation data length was less than 6 years, the accuracy score fluctuated between 0.79 and 0.81. If the training and validation data length was more than 7 years, the accuracy score was improved to between 0.85 and 0.87. Therefore, a sufficiently-long dataset should be needed for training of the MLP model.

Second, we assessed the accurate score and computational efficiency with and without early-stopping (Fig. 10). The computational efficiency was quantified by the ratio of $\frac{t_{\text{total}}-t_{\text{load}}}{t_{\text{load}}}$, where t_{load} represents the time for loading data and t_{total} represents the total time for training and validation in the MLP model. The computational efficiency increased as the ratio decreased. A series of tests were conducted with a set of fractions of X for validation (right panel of Fig. 10) and with varied sizes of mini-batch (left panel of Fig. 10), while keeping other parameters fixed. For the joint tests on fractions of X for validation, fractions were tested at 0.1, 0.2, 0.3, 0.4, and 0.5, while the mini-batch size was kept as 200. For

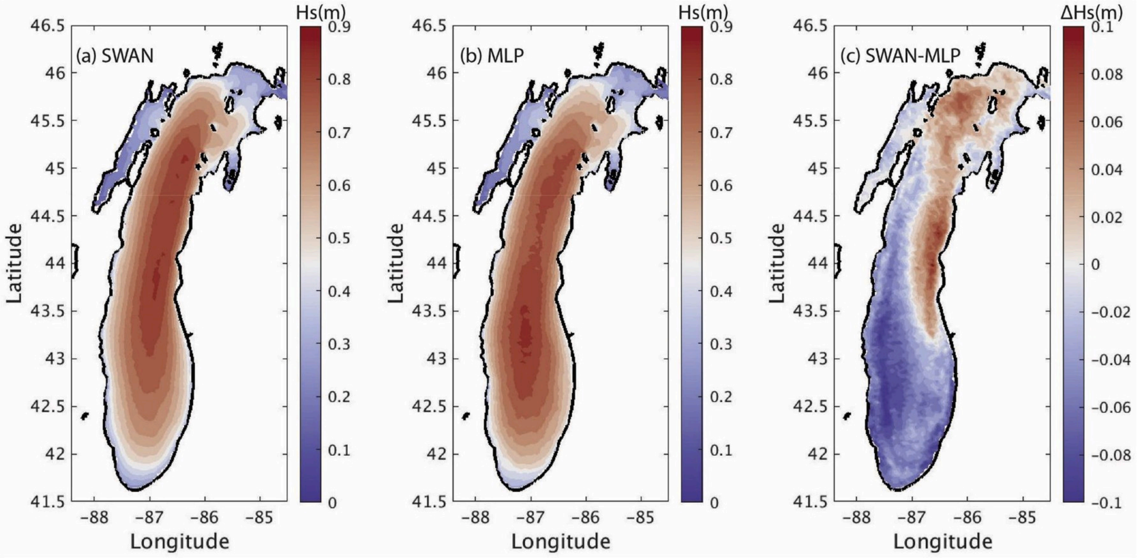


Fig. 7. Distributions of predicted yearly-averaged $\boldsymbol{H_s}$ from (a) the SWAN model and (b) the MLP model in year 2015 as well as (c) the differences in yearly-averaged $\boldsymbol{H_s}$ ($\boldsymbol{H_s}$, **swan** - $\boldsymbol{H_s}$, ***MLP***) (right panel).

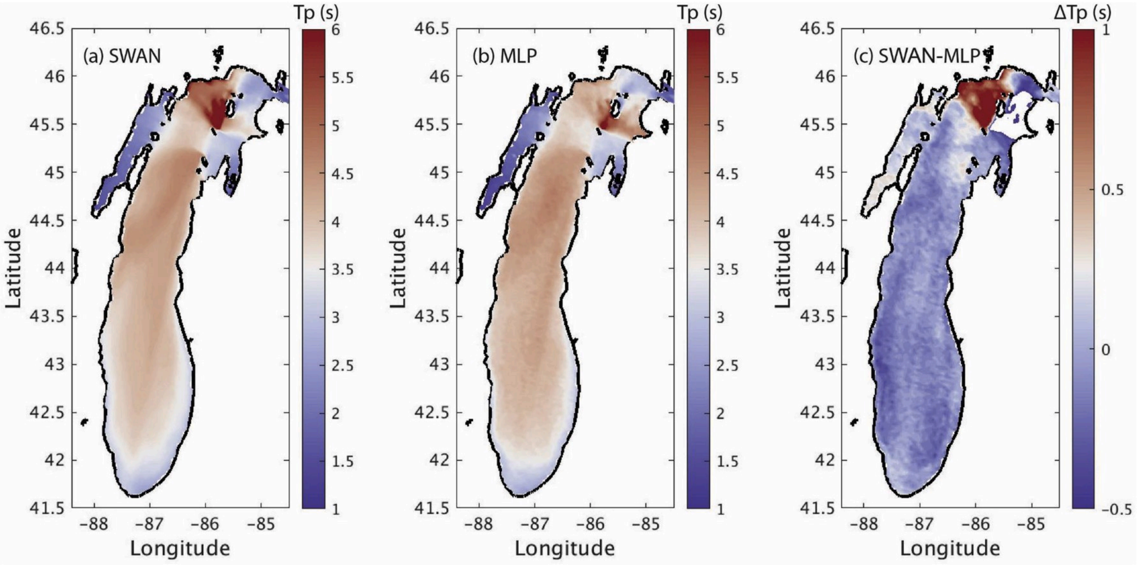


Fig. 8. Distributions of predicted yearly-averaged $\boldsymbol{T_p}$ from (a) the SWAN model and (b) the MLP model in year 2015 as well as (c) the differences in yearly-averaged $\boldsymbol{T_p}$ ($\boldsymbol{T_p}$, **swan** -$\boldsymbol{T_p}$, ***MLP***) (right panel).

the joint test on mini-batch sizes, the mini-batch sizes were 100, 200, 300, 400, and 500. The fraction was set to 0.1 in the joint tests on mini-batch sizes. For the other parameters, activation = 'relu', $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, epsilon = 1e-08, hidden_layer_sizes=(300,300), learning_rate = 'constant', learning_rate_init = 0.001, max_iter = 400, power_t = 0.5, random_state = None, shuffle = True, solver = 'adam', tol = 0.0001.

As shown in Fig. 10, for both fraction tests and mini-batch tests, when the early-stopping was 'false', the accurate score was larger (above 0.85). However, the computational efficiency showed no suggestive results. Judging from the fact that the total time for training and validation are short, in the range of 50–90 s, it was highly suspected that the processing and memory of the computers cast bigger impacts on the computational time ratio than that caused by the early stopping option in the MLP regressor.

Within the fraction tests, when the fraction of X for validation equaled 0.2, the accurate score was the highest; although, neither the accurate score nor the computational efficiency was very sensitive to the fraction. Using multi-year data in 6-h time-resolution for composing X, the model was well-trained even when the fraction reached 0.5. Within the tests on sizes of mini-batch, it was determined that the accurate score was not sensitive to sizes of mini-batch; whereas, the computational time dropped when the size of the mini-batch was over 300. It was found that a mini-batch size of 200 produced higher accurate scores.

Third, we tested the sensitivity from different learning rates of the MLP regressor on the accurate score and computational efficiency (Fig. 11). There are three options: 'constant', 'adaptive', and 'invscaling'. The 'constant' is a constant learning rate given by the initial value and 'adaptive' keeps the rating rate constant to the initial value as long as the training loss keeps decreasing. The 'invscaling' represents gradually decreasing the learning rate at each time step using an inverse scaling exponent of a certain value defined by 'power_t'. Results show that 'adaptive' and 'invscaling' perform slightly better in terms of a higher accurate score and less computational time. However, the

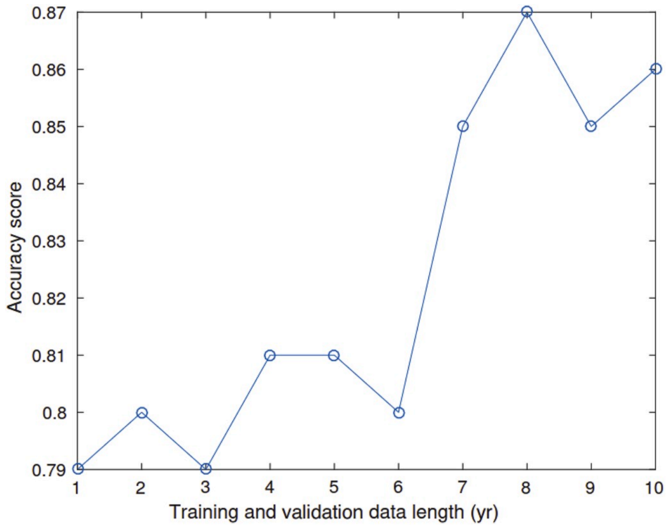


Fig. 9. The accuracy scores of the validation dataset versus the training and validation data length in years.

differences among the results of the three learning rates are not significant, as all the accurate scores are above 0.8 and the total training time of the MLP models are within minutes of one another.

Lastly, we test the stepsize (α) and exponential decay rates for momentum estimates (β_1, β_2) of the MLP regressor on the models performance (Figs. 11 and 12). Following Kingma and Ba (2014), we set $\alpha =$ 0.1, 0.01, 0.001, 0.0001, and 0.00001, $\beta_1 = 0$, 0.5, and 0.9, $\beta_2 = 0.99$, 0.999, and 0.9999. The accurate score is sensitive to α, as α decreases, the accurate score increases. When α is larger than 0.01, the accurate score drops greatly (below 0.8) and the computational time drops as well. With small β_1 values, the accurate score could also be affected. It is found that models with $\alpha \leq 0.01$ and $\beta_1 \geq 0.9$ give higher accurate scores (above 0.85). The modeling results are not sensitive to β_2, as long as $\beta_2 \in [0.99, 1)$.

Except for the intense data requirement and multiple hyperparameters for calibration, the MLP model is found superior to a physics-based wave model on wave forecasting in a number of ways. First, with the trained MLP model, wave forecasting can be greatly accelerated. For example, wave forecasting of H_s and T_p in year 2015 using the MLP model took less than 1 s on a Mac desktop with one single processor, while the physics-based SWAN model took about 4.5 h with the same outputs on an equivalent number of processors in an HPC cluster. The MLP model could forecast wave conditions in a fraction of the time, 1/10,000th to 1/20,000th, to run the physics-based SWAN model. With

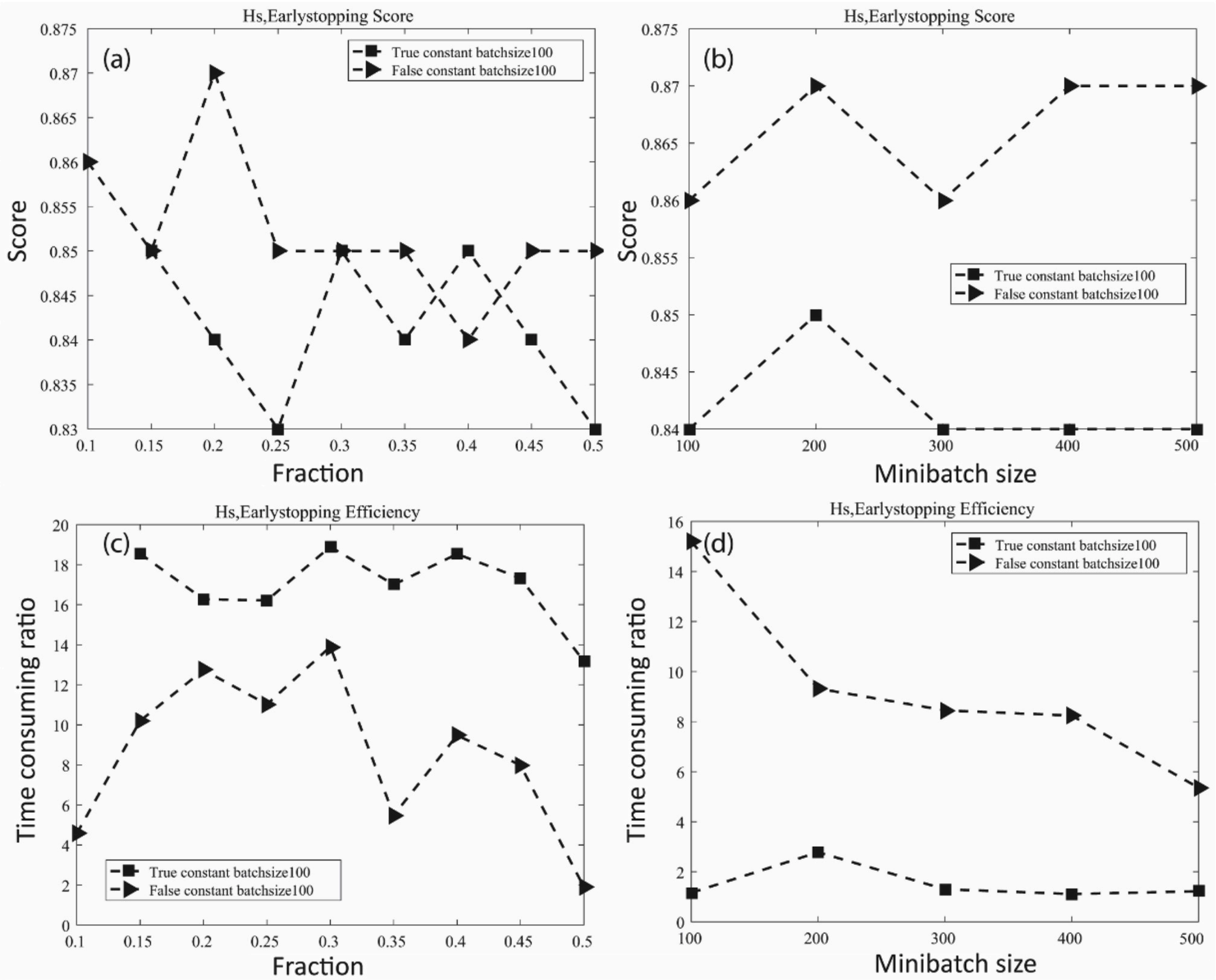


Fig. 10. The accuracy scores and time consuming ratio from testing early stopping (True vs. False) with different fractions of (a,c) validation datasets and with different mini-batch sizes (b,d).

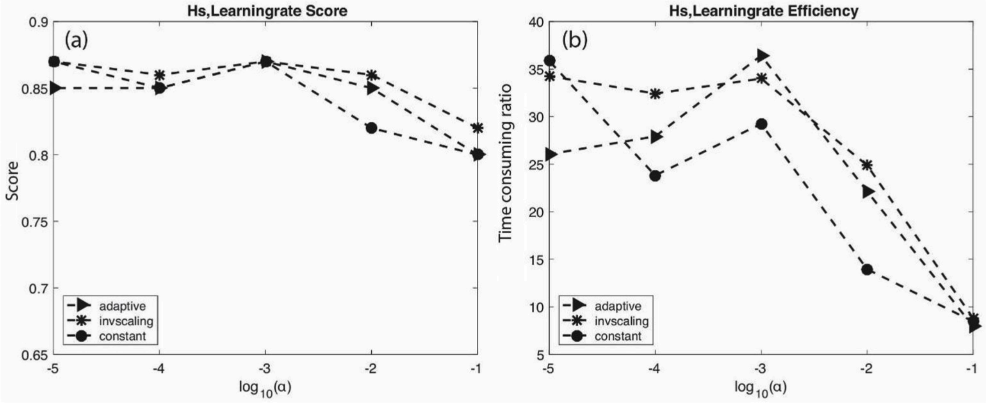


Fig. 11. (a) The accuracy scores and (b) computational time consuming ratio from testing the learning rate with different step sizes ($\boldsymbol{\alpha}$). The learning rate is indicated by different symbols (triangle indicates "adaptive", square indicates "invscaling" and circle indicates "constant"). The x-axis is indicated by $\boldsymbol{log_{10}(\alpha)}$.

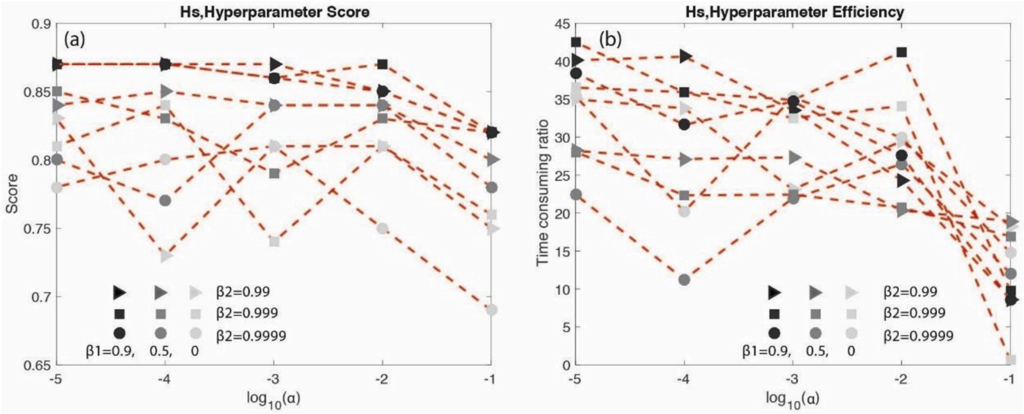


Fig. 12. (a) The accuracy scores and (b) computational time consuming ratio on testing step size ($\boldsymbol{\alpha}$) and hyper-parameters ($\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$). Beta1 is indicated by different marker colors (black indicates 0.9, dark grey indicates 0.5, and light grey indicates 0). $\boldsymbol{\beta_2}$ is indicated by different symbols (triangle indicates 0.99, square indicates 0.999, and circle indicates 0.9999). The x-axis is indicated by $\boldsymbol{log_{10}(\alpha)}$. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

the trained MLP model, massive wave calculations across extremely large spatial and temporal scales could be done in a matter of minutes. Second, the MLP model is easy to use for future wave simulations. To run forward predictions using the trained MLP model, only the matrices of weights, biases, and activation functions are required (DeVries et al., 2017), which can be saved in a small data file. With fewer than 20 lines of Python script, and a small data file, time-dependent wave prediction at any location and time in Lake Michigan can be performed on any computer or device. It is possible to integrate the trained Lake Michigan MLP model into an App for use in a smart phone.

4. Conclusions

A machine learning framework based on an MLP regressor was established for wave forecasting in Lake Michigan. A systematic analysis on the performance of the MLP regressor for a long-term wave characteristics hindcast/forecast was conducted in this study. Different to previous work focusing on ocean-wave conditions, the present study utilized the ML tool in a lacustrine environment and ice-cover was, for the first time, considered in the application of an MLP regressor on wind-generated wave modeling in an enclosed and deep lake. The ML tool used wind field and ice coverage as inputs ('features' in ML terms) and output wave characteristics ('labels' in ML terms) including significant wave height and peak wave period in the entire lake. The model was trained using the simulated H_s and T_p data from a physics-based SWAN wave model for the period 2005–2014. The trained MLP model was computationally efficient and capable of predicting characteristic wave conditions (H_s and T_p), with accuracy comparable to that of the SWAN wave model in the validation dataset. Wave forecasting in 2015 showed that the correlations between the MLP model and the SWAN model were 0.92 for H_s and 0.79 for T_p, respectively. This study also showed that the MLP regressor had the ability to model wave period (e.g. T_p) with results comparable to SWAN modelling.

This study provided guidance for wave forecast/hindcast in high latitudes utilizing an MLP regressor. Sensitivity tests on hyper-parameters and regularization techniques were performed for the development and validation of the MLP model. The robustness of the MLP model was extensively demonstrated and a suggestive table for selection of the MLP model parameters was provided. The 'adam' solver was suggested for case studies with intense data similar to the work presented in this paper. Once the training and validation dataset is sufficiently long (>7 years), the MLP model was quite robust in terms of prediction accuracy (for 1-year forecasting). Results showed that 'learning rate' set as 'adaptive' and 'invscaling' performed slightly better in terms of higher accuracy scores and computational efficiency than when set as 'constant'. It was suggested that activity function be set as 'Relu' for faster computation. The option 'Early-stopping' in certain circumstances could save computational time when set to 'true', but it was determined that higher accuracy scores were achieved when it was set to 'false'. It was also shown that the MLP regressor gave higher prediction scores (above 0.85), with mini-batch $\geq$200, stepsize $\leq$ 0.01, and an exponential decay rate for the first momentum $\geq$0.9, along with an exponential decay rate for the second momentum within the range of $[0.99, 1)$.

The trained MLP model could act as an efficient wave forecasting system for Lake Michigan. Wave forecasting of characteristic wave conditions for one year took less than 1 s using the MLP model. By comparing the computational efficiency of the MLP model and the

SWAN model for wave forecasting, it was determined that the computational time of the trained MLP model is a fraction (1/20,000th to 1/10,000th) of that of the physics-based wave model SWAN.

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.

CRediT authorship contribution statement

Xi Feng: Writing - original draft, Conceptualization, Visualization, Investigation. **Gangfeng Ma:** Methodology, Software, Supervision. **Shih-Feng Su:** Conceptualization, Methodology. **Chenfu Huang:** Data curation. **Maura K. Boswell:** Writing - review & editing. **Pengfei Xue:** Writing - review & editing.

Acknowledgments

We would like to give our acknowledgement to the National Natural Science Foundation of China [No. 51709091]; the Natural Science Foundation of Jiangsu Province [No. BK20170874]; and the Fundamental Research Funds for the Central Universities [No. 2017B005] for the funding support. This work was also supported by the Michigan Sea Grant College Program, project number (R/CGLH-7), under [NA18OAR4170102], from NOAA National Sea Grant, U.S. Department of Commerce and funds from the State of Michigan. This is Contribution No. 73 of the Great Lakes Research Center at Michigan Tech.

References

Akpinar, Adem, Bingolbali, B., Van Vledder, G.P., 2016. Wind and wave characteristics in the Black Sea based on the SWAN wave model forced with the CFSR winds. Ocean Eng. 126, 276–298. Nov.1.

Allahdadi, M. Nabi, Gunawan, Budi, Lai, Jonathan, He, Ruoying, Vincent, S. Neary, 2019. Development and validation of a regional-scale high-resolution unstructured model for wave energy resource characterization along the US East Coast. Renew. Energy 136, 500–511.

Ahmad, S., Kalra, A., Stephen, H., 2010. Estimating soil moisture using remote sensing data: a machine learning approach. Adv. Water Resour. 33, 69–80.

Anderson, J.D., Wu, C.H., Schwab, D.J., 2015. Wave climatology in the apostle islands, lake superior. J. Geophys. Res. Oceans 120, 4869–4890.

Assel, R., Cronk, K., Norton, D., 2003. Recent trends in Laurentian Great lakes ice cover. Climatic Change 57 (1–2), 185–204.

Assel, R.A., 2005. Great Lakes Ice Cover Climatology Update: Winters 2003, 2004, and 2005. NOAA Technical Memorandum GLERL-135. NOAA,Great Lakes Environmental Research Laboratory, Ann Arbor, MI.

Assel, R.A., Wang, J., Cites, A.H., Bai, X., 2013. Analysis of Great Lakes Ice Cover Climatology: Winters 2006-2011, NOAA Technical Memorandum GLERL-157. NOAA Great Lakes Environmental Research Laboratory, Ann Arbor, MI.

Bennington, V., McKinley, G.A., Kimura, N., Wu, C.H., 2010. General circulation of Lake superior: mean, variability and trends from 1979 to 2006. J. Geophys. Res. Oceans 115, C12015. https://doi.org/10.1029/2010JC006261.

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

Chawla, A., Spindler, D.M., Tolman, H.L., 2013. Validation of a thirty year wave hindcast using the Climate Forecast System Reanalysis winds. Ocean Model. 70, 189–206.

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

DeVries, P.M., Thompson, T.B., Meade, B.J., 2017. Enabling large-scale viscoelastic calculations via neural network acceleration. Geophys. Res. Lett. 44, 2662–2669.

Etemad-Shahidi, A., Bonakdar, L., 2009. Design of rubble-mound breakwaters using M5 machine learning method. Appl. Ocean Res. 1, 197–201.

Etemad-Shahidi, A., Yasa, R., Kazeminezhad, M.H., 2011. Prediction of wave-induced scour depth under submarine pipelines using machine learning approach. Appl. Ocean Res. 33, 54–59.

Erikson, L.H., Hegermiller, C.A., Barnard, P.L., Ruggiero, P., van Ormondt, M., 2015. Projected wave conditions in the Eastern North Pacific under the influence of two CMIP5 climate scenarios. Ocean Model. 1–15.

Goda, Y., 2010. Random Seas and Design of Maritime Structures, third ed. World Scientific.

Gunaydin, K., 2008. The estimation of monthly mean significant wave heights by using artificial neural network and regression methods. Ocean Eng. 35, 1406–1415.

Hong, W.-C., 2008. Rainfall forecasting by technological machine learning models. Appl. Math. Comput. 200, 41–57.

Hubertz, J.M., Driver, D.B., Reinhard, R.D., 1991. Wind waves on the Great lakes: a 32 year hindcast. J. Coast Res. 7, 945–967.

James, S.C., Zhang, Y., O'Donncha, F., 2018. A machine learning framework to forecast wave conditions. Coast. Eng. 137, 1–10.

Kingma, Diederik, Ba, Jimmy, 2014. Adam: A Method for Stochastic Optimization arXiv preprint arXiv:1412. 6980.

Komen, G.J., Cavaleri, I., Donelan, M., Hasselmann, K., Hasselmann, S., Janssen, P.A.E. M., 1994. Dynamics and Modeling of Ocean Waves. Cambridge University Press.

Kukulka, Tobias, et al., 2017. Surface wave dynamics in Delaware Bay and its adjacent coastal shelf. J. Geophys. Res.: Oceans 122 (11), 8683–8706.

Kutupoğlu, Volkan, EmreÇakmak, Recep, Akpınar, Adem, Phvan Vledder, Gerbrant, 2018. Setup and evaluation of a SWAN wind wave model for the Sea of Marmara. Ocean Eng. 165 (Oct.1), 450–464.

Lary, D.J., Alavi, A.H., Gandomi, A.H., Walker, A.L., 2016. Machine learning in geosciences and remote sensing. Geosci. Front. 7, 3–10.

Lee, T.-L., 2006. Neural network prediction of a storm surge. Ocean Eng. 33, 483–494.

Li, Jiangxia, Chen, Yongping, Pan, Shunqi, Pan, Yi, Fang, Jiayu, Derrick, M.A., 2016. Sowa. "Estimation of mean and extreme waves in the East China seas. Appl. Ocean Res. 56, 35–47.

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.

Mori, N., Yasuda, T., Mase, H., Tom, T., Oku, Y., 2010. Projection of extreme wave climate change under global warming. Hydrol. Res. Lett. 4, 15–19.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machine Learning.

Niroomandi, A., Ma, G., Ye, X., Lou, S., Xue, P., 2018. Extreme value analysis of wave climate in Chesapeake Bay. Ocean Eng. 159, 22–36.

O'Donncha, Fearghal, Zhang, Yushan, Chen, Bei, James, Scott C., 2018. An integrated framework that combines machine learning and numerical models to improve wave-condition forecasts. J. Mar. Syst. 186, 29–36.

O'Donncha, Fearghal, Zhang, Yushan, Chen, Bei, James, Scott C., 2019. Ensemble model aggregation using a computationally lightweight machine-learning model to forecast ocean waves. J. Mar. Syst. 199, 103206.

Rasouli, K., Hsieh, W.W., Cannon, A.J., 2012. Daily streamflow forecasting by machine learning methods with weather and climate inputs. J. Hydrol. 414–415, 284–293.

Saha, S., Moorthi, S., Pan, H.-L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., et al., 2010. The NCEP climate forecast system reanalysis. Bull. Am. Meteorol. Soc. 91, 1015–1057.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H.-Y., Iredell, M., et al., 2014. The NCEP climate forecast system version 2. J. Clim. 27, 2185–2208.

Shi, Jian, et al., 2019. A 39-year high resolution wave hindcast for the Chinese coast: model validation and wave climate analysis. Ocean Eng. 83 (JUL.1), 224–235.

Soares, Guedes C., Scotto, M.G., 2007. Application of the r-order statistics for long-term predictions of significant wave heights. Coast Eng. 51, 387–394.

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

Tsai, C.-P., Lin, C., Shen, J.-N., 2002. Neural network for wave forecasting among multi-stations. Ocean Eng. 29, 1683–1695.