A machine learning framework to forecast wave conditions

Scott C. James [a,*], Yushan Zhang [b], Fearghal O'Donncha [c]

[a] Baylor University, Departments of Geosciences and Mechanical Engineering, One Bear Place #97354, Waco, TX 76798-2534, USA
[b] University of Notre Dame, Department of Chemical and Biomolecular Engineering, Notre Dame, IN 46556-5637, USA
[c] IBM Research, Dublin, Ireland

ARTICLE INFO

ABSTRACT

A machine learning framework is developed to estimate ocean-wave conditions. By supervised training of machine learning models on many thousands of iterations of a physics-based wave model, accurate representations of significant wave heights and period can be used to predict ocean conditions. A model of Monterey Bay was used as the example test site; it was forced by measured wave conditions, ocean-current nowcasts, and reported winds. These input data along with model outputs of spatially variable wave heights and characteristic period were aggregated into supervised learning training and test data sets, which were supplied to machine learning models. These machine learning models replicated wave heights from the physics-based model with a root-mean-squared error of 9 cm and correctly identify over 90% of the characteristic periods for the test-data sets. Impressively, transforming model inputs to outputs through matrix operations requires only a fraction ($< 1/1,000^{\text{th}}$) of the computation time compared to forecasting with the physics-based model.

1. Introduction

There are myriad reasons why predicting wave conditions is important to the economy. Surfers aside, there are fundamental reasons why knowledge of wave conditions for the next couple of days is important. For example, shipping routes can be optimized by avoiding rough seas thereby reducing shipping times. Another industry that benefits from knowledge of wave conditions is the $160B (2014) aquaculture industry (FAO, 2016), which could optimize harvesting operations accordingly. Knowledge of littoral conditions is critical to military and amphibious operations by Navy and Marine Corps teams. Also, predicting the energy production from renewable energy sources is critical to maintaining a stable electrical grid because many renewable energy sources (e.g., solar, wind, tidal, wave, etc.) are intermittent. For deeper market penetration of renewable energies, combinations of increased energy storage and improved energy-generation predictions will be required. The US Department of Energy has recently invested in the design, permitting, and construction of an open-water, grid-connected national Wave Energy Test Facility at Oregon State University (US DOE, 2016). Given that America's technically recoverable wave-energy resource is up to 1,230 TW-hr (EPRI, 2011), there is a strong interest in developing this renewable resource (Ocean Energy Systems, 2016). Commercialization and deployment of wave-energy technologies will require not only addressing permitting and regulatory matters, but overcoming technological challenges, one of which is being able to provide an accurate prediction of energy generation. A requirement for any forecast is that an appropriately representative model be developed, calibrated, and validated. Moreover, this model must be able to run extremely fast and to incorporate relevant forecast data into its predictions. A machine learning framework for this capability is developed here.

Because wave models can be computationally expensive, a new approach with machine learning (Goodfellow et al., 2016; LeCun et al., 2015; Schmidhuber, 2015) is developed here. The goal of this approach is to train machine learning models on many realizations of a physics-based wave model forced by historical atmospheric and sea states to accurately represent wave conditions (specifically, significant wave heights and characteristic period). Predicting these wave conditions at locations corresponding to a (potential) wave-energy-converter (WEC) array facilitates accurate power-production forecasts. Given the recent development of a wave-energy-resource classification system (Haas et al., 2017), if the wave conditions at a particular location can be predicted, the power potential for a hypothetical WEC array can be estimated.

Computational expense is often a major limitation of real-time forecasting systems (DeVries et al., 2017; Mallet et al., 2009). Here, we apply machine learning techniques to predict wave conditions with the goal of replacing a computationally intensive physics-based model by

---

* Corresponding author.
  E-mail address: sc_james@baylor.edu (S.C. James).

straightforward multiplication of an input vector by mapping matrices resulting from the trained machine learning models. Because matrix multiplication is an exceedingly rapid operation, the end result is a machine learning technique that can predict wave conditions with comparable accuracy to a physics-based model for a fraction of the computational cost. While machine learning has been used to predict wave conditions (Peres et al., 2015; Makarynskyy, 2004; Etemad-Shahidi and Mahjoobi, 2009; Mahjoobi and Etemad-Shahidi, 2008; Browne et al., 2006, 2007), it has not been used in the context of a surrogate model as defined below.

One of the challenges for machine learning applications is their enormous appetite for data. It is the exception more than the rule that a machine learning approach has what is considered an optimal amount of data. However, when developing a machine learning surrogate for a physics-based model, there is the luxury of being able to run the physics-based model as many times as necessary to develop a sufficiently large data set to train the machine learning model. Here, we define a surrogate model (Razavi et al., 2012) as a data-driven technique to empirically approximate the response surface of a physics-based model. These have alternately been called "metamodels" (Blanning, 1975; Kleijnen, 2009), "model emulators" (O'Hagan, 2006), and "proxy models" (Bieker et al., 2007). To assemble the training dataset required to develop a robust machine learning model, the inputs and outputs of many thousands of wave model runs were accumulated into a suitably large set of input vectors.

2. Wave modeling

2.1. Numerical model

The Simulating WAves Nearshore (SWAN) V 41.62 FORTRAN code is the industry-standard wave-modeling tool developed at the Delft University of Technology that computes wave fields in coastal waters forced by wave conditions on the domain boundaries, ocean currents, and winds (The SWAN Team, 2006). SWAN models the energy contained in waves as they travel over the ocean and disperse at the shore. Specifically, information about the sea surface is contained in the wave-variance spectrum, or energy density $E(\sigma, \theta)$, and this wave energy is distributed over wave frequencies (as observed in an inertial frame of reference moving with the current velocity) with propagation directions normal to wave crests of each spectral component.

Action density is defined as $N = E/\sigma$, which is conserved during propagation along the wave characteristic in the presence of ambient current. Evolution of $N(x, y, t; \sigma, \theta)$ in space, x, y, and time, t, is governed by the action balance equation (Komen et al., 1996; Mei et al., 1989):

$$\frac{\partial N}{\partial t} + \left(\frac{\partial c_x N}{\partial x} + \frac{\partial c_y N}{\partial y} \right) + \left(\frac{\partial c_\sigma N}{\partial \sigma} + \frac{\partial c_\theta N}{\partial \theta} \right) = \frac{S_{\text{tot}}}{\sigma}. \tag{1}$$

The left-hand side represents the kinematic component of the equation. The second term (parenthetical) denotes the propagation of wave energy in a two-dimensional Cartesian space where c is wave celerity. The third term represents the effect of shifting of the radian frequency due to variations in water depth and mean current. The fourth term expresses depth- and current-induced refractions. The quantities c_σ and c_θ are the propagation speeds in spectral space (σ, θ). The right-hand side represents the spatio-temporally variable sources and sinks of all physical processes that generate, dissipate, or redistribute wave energy (i.e., wave growth by wind, nonlinear transfer of energy through three- and four-wave interactions, and wave decay due to white-capping, bottom friction, and depth-induced wave breaking).

Haas et al. (2017) define the wave-energy resource as a function of the significant wave height, H_s and peak wave period, T. This information can be used to compute the wave power density. Hence, estimates of peak period T and, in particular, H_s because J is quadratically related to wave height, are necessary to predict wave energy potential.

2.2. Model verification

The coastal ocean presents a complex modeling challenge, intimately connected as it is to both the deep ocean and the atmosphere (Song and Haidvogel, 1994). Uncertainties in wave forecasting emanate from the mathematical representation of the system, numerical approximations, and uncertain and incomplete data sets. Studies demonstrate that the greatest sources of uncertainty in operational wave forecasting are the model input data. This study simulates wave conditions subject to real forcing conditions at a case-study site, Monterey Bay, California. As summarized in Table 1, the wave model was driven by available NOAA wave-condition data, archived current nowcasts from the Central and Northern California Ocean Observing System (CeNCOOS) (Patterson et al., 2012), and post-processed (i.e., data subject to quality-assurance procedures) wind data from The Weather Company (2017).

Before developing a machine-learning surrogate for the physics-based SWAN model, it is important to demonstrate that SWAN can accurately replicate wave conditions in Monterey Bay so that a training data set can be developed for the machine learning models. The SWAN model validated by Chang et al (2016) was used in this effort because it has a demonstrated track record for accurately simulating wave conditions in Monterey Bay. The bathymetric data shown in Fig. 1 were obtained from the NOAA National Geophysical Data Center. The horizontal resolution in this SWAN model was 0.001°.

Because SWAN discretizes wave frequencies in its calculations, only a user-defined number of discrete T values can be returned by a simulation. Specifically, because this effort is relevant to forecasts for wave-energy converters, we consider the peak wave period (RTP in the SWAN model) because this is used in the calculation for wave power (Cahill and Lewis, 2014). To this end, when building the model the user specifies the minimum, ϕ_1, maximum ϕ_N, and number of discrete frequencies, N, which are logarithmically distributed as (The SWAN Team, 2006):

$$\phi_i = \phi_{i-1} + \phi_{i-1} \left(\frac{\phi_N}{\phi_1} - 1 \right)^{\frac{1}{N-1}}. \tag{2}$$

Note that the logarithmic distribution yields smaller increments between periods for larger T, which is most relevant for capturing the effects of long-period waves – those most important for energy generation. For these simulations, $\phi_1 = 0.042$ Hz ($T_1 = 23.8$ s) and $\phi_N = 1$ Hz ($T_N = 1$ s), and N = 24 discrete T values were specified (Chang et al., 2016). However, only 11 distinct T values were ever calculated by SWAN

Table 1
Data sources.

Data	Source	URL	Resolution
Wave conditions (H_s, T, D)	NDBC[c,a] Buoy 46042	http://www.ndbc.noaa.gov/ station_page.php?station = 46042	36°47′29″ N 122°27′6″ W
Wave conditions (H_s, T, D)	WAVE WATCH III ENP NCEP[b]	http://nomads.ncep.noaa.gov: 9090/dods/wave/enp	0.25°
Ocean currents	ROMS COPS[d]	http://west.rssoffice.com:8080/ thredds/catalog/roms/CA3000m-forecast/catalog.html	3 km
Winds	TWC[d]	https://api.weather.com/	User defined
Bathymetry	NOAA NGDC[e]	https://www.ngdc.noaa.gov/mgg/ bathymetry/hydro.html	0.001°

[a] National Data Buoy Center.
[b] Eastern North Pacific National Centers for Environmental Prediction.
[c] Regional Ocean Modeling System Cooperative Ocean Prediction System.
[d] The Weather Company.
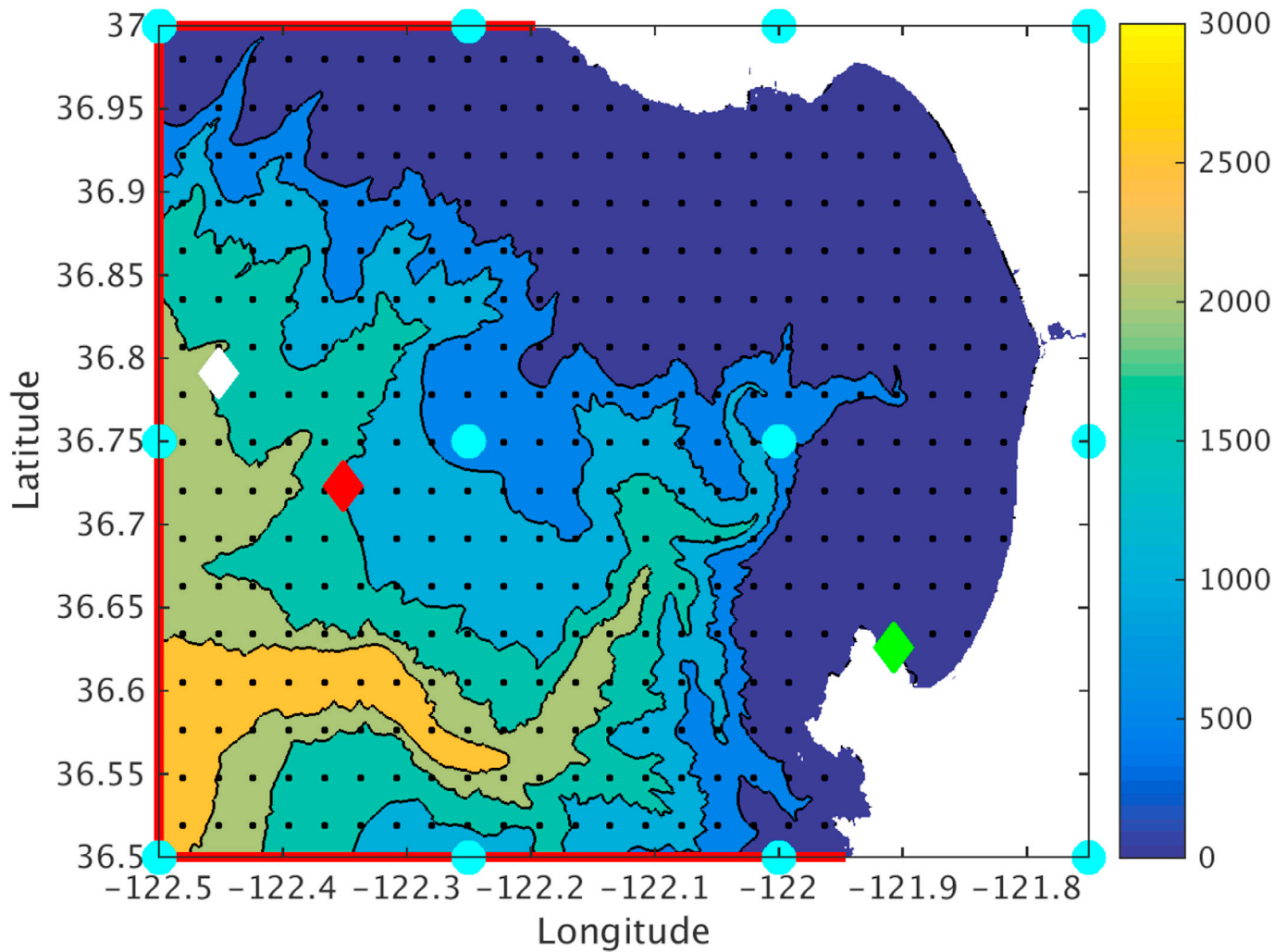[e] National Geophysical Data Center.

Fig. 1. SWAN model domain with color indicating bathymetric depth. The three buoys used to verify the model are indicated with the symbols where the white diamond is Buoy 46042, the red diamond is Buoy 46114, and the green diamond is Buoy 46240. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

because NOAA buoy data ranged from 3.98 to 15.40 s, so henceforth, N = 11. Throughout most of the model domain, SWAN calculated a single "characteristic T" with only minimal variation observed in shallow waters comprising about 2% of the model cells. Hence, we define a characteristic T for each run of the SWAN model because this is an important wave characteristic for estimating wave energy.

Before developing the machine learning data set, it is important to verify that SWAN can simulate wave characteristics with sufficient accuracy. For this model-verification exercise, inputs forcing the SWAN model comprised wave conditions (defined along the SWAN model boundaries indicated in red in Fig. 1) from the NOAA National Data Buoy Center (Buoy 46042), ocean currents from CeNCOOS (Patterson et al., 2012) (357 u and v currents supplied at the 3-km-spaced small, black circles in Fig. 1), and wind data from The Weather Company (2017) extracted at 12 locations spaced 0.25° apart corresponding to the turquoise circles in Fig. 1. The regional wave model is the WAVEWATCH III (Tolman et al, 2009) simulation of the Eastern North Pacific discretized at 0.25° (turquoise circles in Fig. 1) and its simulations are representative of the typical accuracies in a regional model. Ocean currents are from the Monterey Bay ocean-forecasting system based on the 3-km-resolution Regional Ocean Modeling System (ROMS) (Integrated Ocean Observing System, 2017). ROMS atmospheric forcing is derived from the Coupled Ocean/Atmosphere Mesoscale Prediction System atmospheric model

(Hodur, 1997). Akin to the SWAN model, ROMS wave and tidal forcing are specified at the three lateral boundaries of the outermost domain using information from the global tidal model (Dushaw et al., 1997). Wind speeds were extracted at 0.25° spacing from a TWC application programming interface (API). TWC provides information on a variety of meteorological conditions, forecasts, alerts, and historical data, which can be extracted either directly from the TWC API or through the IBM Bluemix platform. Hourly forecast data out to fifteen days are available along with historical cleansed data for the past 30 years.

Six days of NOAA wave data, ROMS ocean currents, and TWC winds were assembled into *steady-state* SWAN model runs at 3-h intervals. Fig. 2 is an example SWAN-simulated H_s field showing waves entering the domain from the north-northwest and diffracting around the northern coast of Monterey Bay. Fig. 3 compares NOAA wave-condition data (significant wave height, H_s, top row; wave period, T, middle row; wave direction, D, bottom row) for the three buoys in the Monterey Bay area (red curves) to WAVEWATCH III forecasts (black symbols) at that model's grid point nearest each buoy. Given the vast extent of the relatively coarse WAVEWATCH III model domain (0.25° resolution for the entire Eastern North Pacific compared to 0.001° of the SWAN model), it is not surprising that there was a degree of local mismatch. Moreover, the location of NOAA Buoy 46240 (green diamond in Fig. 1), which is sheltered from incoming westward waves, resulted in a notable
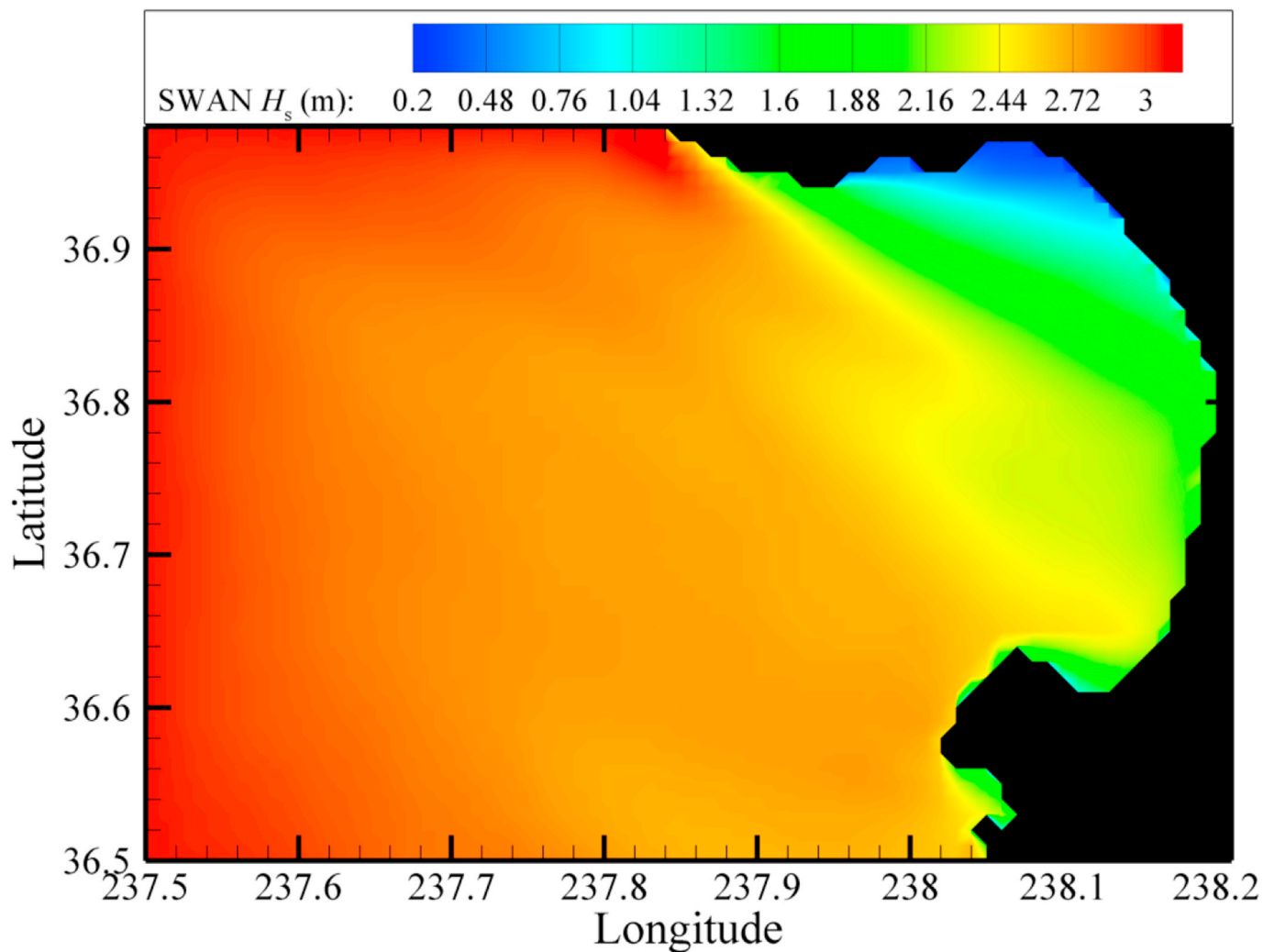
Fig. 2. Representative SWAN H_s field.

discrepancy between WAVEWATCH III-simulated and measured wave conditions. This was expected because the nearest WAVEWATCH III model node is 15 km away at the turquoise circle to the northwest of the green diamond in Fig. 1. Blue symbols are SWAN-simulated wave conditions when NOAA wave conditions from Buoy 46042 were supplied to the SWAN model as boundary conditions. SWAN did a good job of capturing the effects of bathymetry and coastline on the wave conditions at Buoy 46240 given that it was forced by the wave conditions displayed as the black curves (NOAA data) in the left column of Fig. 3. Overall, SWAN was able to simulate wave conditions more accurately than the WAVEWATCH III model. Root-mean-squared errors (RMSEs) are listed in Table 2. Bidlot et al. (2002) note that 40- to 60-cm RMSEs are typical for H_s. The simulations from the SWAN model forced by NOAA wave-condition data yielded an appropriate (RMSE < 50 cm) match to available NOAA data indicating that the SWAN model is acceptable for developing the machine learning model training data set.

For production runs, the SWAN model resolution was coarsened from $0.001°$ to $0.01°$; this yielding 3,104 active nodes in the domain. This model could be run in under 40 s as compared with the refined model, which took minutes to calculate and resulted in 224,163 active nodes. Although solving the refined SWAN model is only a matter of computational resource, the analysis of the spatial covariance indicates that $0.001°$ resolution was not necessary to develop the framework outlined here; neither was a wave field comprising 224,163 H_s values.

3. Machine learning

Two different supervised machine learning models were used to perform two different tasks: regression analysis for wave height and classification analysis for characteristic period. The multi-layer perceptron (MLP) conceptual model used to replicate H_s is loosely based on the anatomy of the brain. Such an artificial neural network is composed of densely interconnected information-processing nodes organized into layers. The connections between nodes are assigned "weights," which determine how much a given node's output will contribute to the next node's computation. During training, where the network is presented with examples of the computation it is learning to perform (i.e., SWAN model runs), those weights are optimized until the output of the network's last layer consistently approximates the result of the training data set (in this case, wave heights). A support vector machine (SVM) classification analysis constructs hyperplanes (planes in high-dimensional space) that divide the training data set into labeled groups (in this case, characteristic Ts).

3.1. Background

Machine learning has shown enormous potential for pattern recognition in large data sets. Consider that a physics-based model like SWAN acts as a nonlinear function that transforms inputs (wave-characteristics boundary conditions and the spatially variable ocean currents and wind
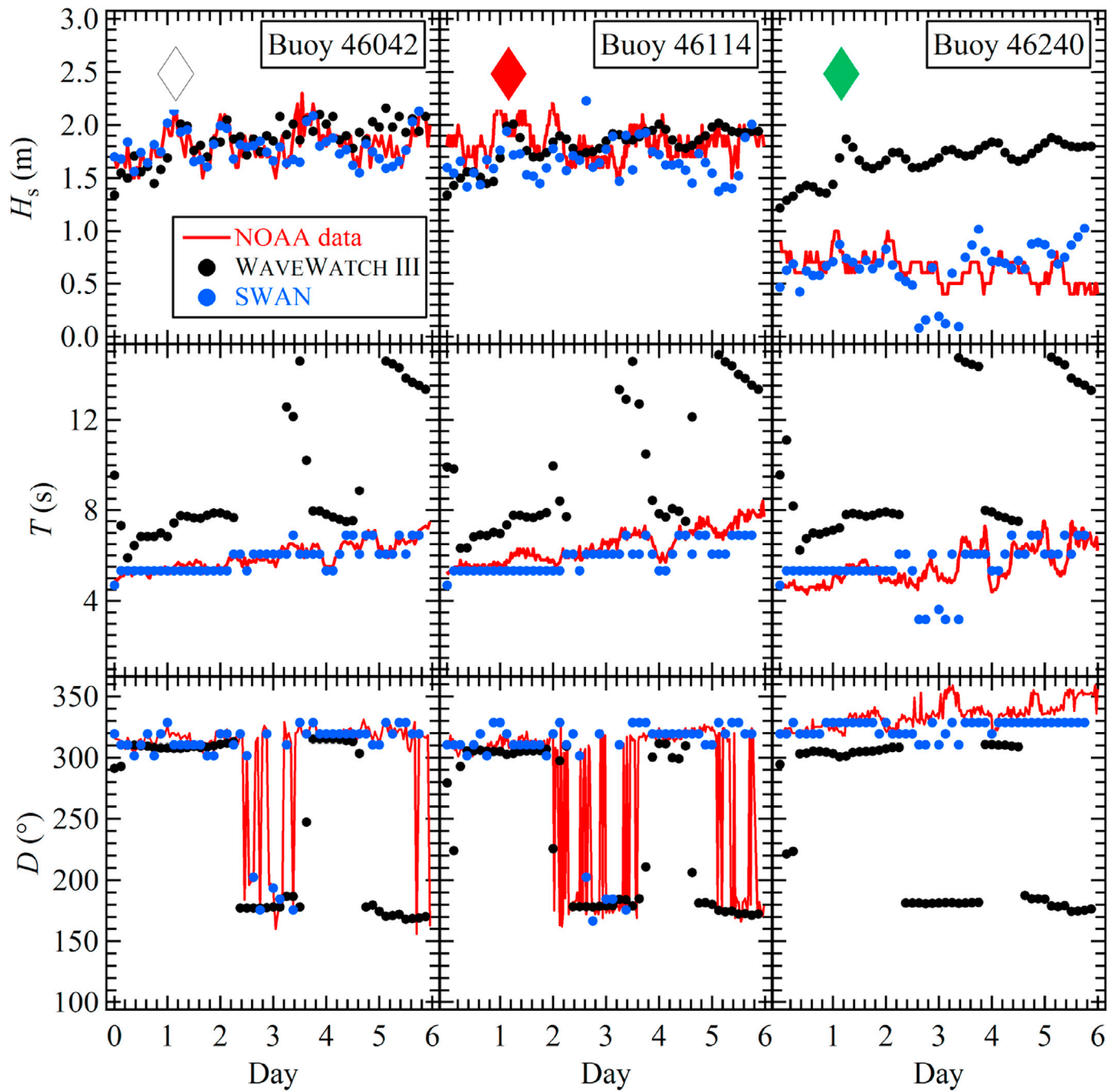
Fig. 3. Comparison of measured and simulated wave conditions at the three NOAA buoys.

Table 2
RMSEs between NOAA wave-condition data and WAVE WATCH III simulations at the grid point nearest to the indicated buoy and SWAN simulations when forced by data from Buoy 46042.

Model	Buoy 46042			Buoy 46114			Buoy 46240		
	H_s (cm)	T (s)	D (°)	H_s (cm)	T (s)	D (°)	H_s (cm)	T (s)	D (°)
WAVE WATCH III	67	3.61	80.3	62	3.29	74.5	162	4.08	103.7
SWAN	50	0.57	62.3	41	0.45	73.9	42	1.06	12.0

speeds) to outputs (spatially variable H_s and characteristic T). These data and corresponding simulations can be assembled into an input vector, $\underline{x}$, and an output vector, $\underline{y}$, respectively.

Because the goal of this effort was to develop a machine learning framework to act as a surrogate for the SWAN model, the nonlinear function mapping inputs to the best representation of outputs, $\widehat{\underline{y}}$, was sought:

$$g(\underline{\mathbf{x}};\underline{\boldsymbol{\Theta}}) = \widehat{\underline{\mathbf{y}}}. \tag{3}$$

A sufficiently trained machine learning model yields a mapping matrix, $\underline{\boldsymbol{\Theta}}$, that can act as a surrogate for the SWAN model. This facilitates sidestepping of the SWAN model by replacing the solution of the partial differential equation with the data-driven machine learning model composed of the vector-matrix operations encapsulated in (3).

The Python toolkit SciKit-Learn (Pedregosa et al., 2011) was used to access high-level programming interfaces to machine learning libraries and to cross validate results. Two distinct machine learning models were implemented here: an MLP model for H_s and an SVM model for characteristic T. Two different approaches were undertaken because the discrete characteristic T values were more accurately represented by an SVM model than an MLP model (more on this later).

3.1.1. The multi-layer perceptron model

An MLP model is organized in sequential layers made up of interconnected neurons. As illustrated in Fig. 4, the value of neuron n in hidden layer ℓ is calculated as:

$$a_n^{(\ell)} = f\left(\sum_{k=1}^{\mathcal{N}_{\ell-1}} w_{k,n}^{(\ell)} a_k^{(\ell-1)} + b_n^{(\ell)} \right), \tag{4}$$

where f is the activation function, $\mathcal{N}_{\ell-1}$ is the number of neurons in layer $\ell-1$, $w_{k,n}^{(\ell)}$ is the weight projecting from node k in layer $\ell-1$ to node n in

layer ℓ, $a_k^{(\ell-1)}$ is the activation of neuron k in hidden layer $\ell-1$, and $b_n^{(\ell)}$ is the bias added to hidden layer ℓ contributing to the subsequent layer. The activation function selected for this application was the rectified linear unit (ReLU) (Nair and Hinton, 2010):

$$f(z) = \max(0, z). \tag{5}$$

A loss function was defined in terms of the squared error between the SWAN predictions and the machine-learning equivalents plus a regularization contribution:

$$L = \frac{1}{2}\sum_{k=1}^{m} ||\underline{\mathbf{y}}^{(k)} - \widehat{\underline{\mathbf{y}}}^{(k)}||_2^2 + \alpha ||\underline{\boldsymbol{\Theta}}||_2^2, \tag{6}$$

where the $||\cdot||_2$ indicates the L_2 norm. The regularization term penalizes complex models by enforcing weight decay, which prevents the magnitude of the weight vector from growing too large because large weights can lead to overfitting where, euphemistically, the machine learning model "hallucinates" patterns in the data set (Goodfellow et al., 2016).

By minimizing the loss function, the supervised machine learning algorithm identifies the $\underline{\boldsymbol{\Theta}}$ that yields $\widehat{\underline{\mathbf{y}}} \approx \underline{\mathbf{y}}$. As shown in Fig. 4, a machine learning model transforms an input vector (layer) to an output layer through a number of hidden layers. The machine learning model is trained on a data set to establish the weights parameterizing the space of nonlinear functions mapping from $\underline{\mathbf{x}}$ to $\underline{\mathbf{y}}$. Of course, a large training data set is required to develop a robust machine learning model; one luxury of
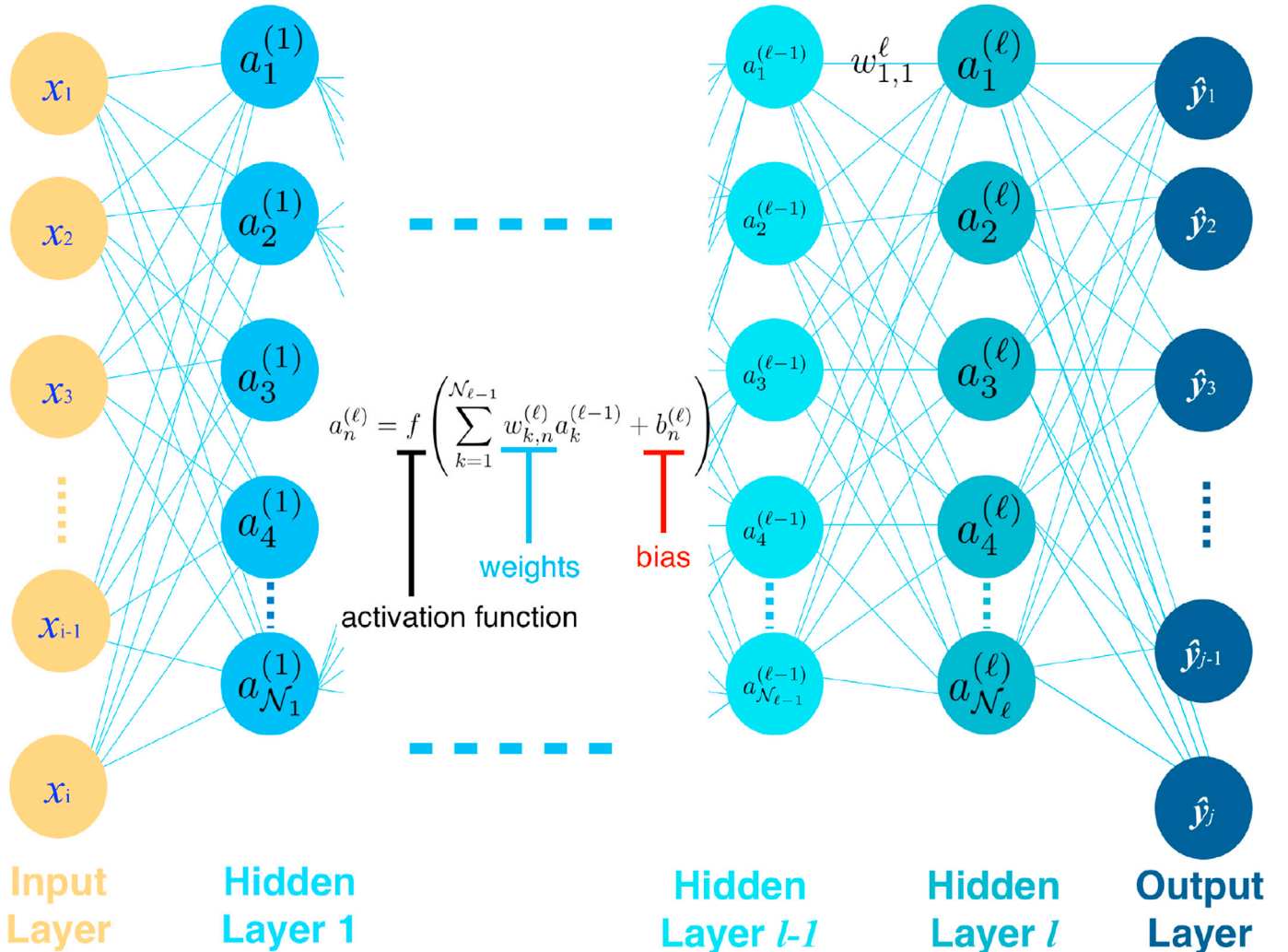

Fig. 4. Schematic of an MLP machine learning network.

the approach developed here is that such a data set is straightforward to assemble by completing many thousands of SWAN model runs to accumulate many input vectors, $\underline{\mathbf{x}}^{(m)}$, into an $(m \times i)$ design matrix, $\underline{\mathbf{X}}$, which when acted upon by functions representing the hidden layers, that is, the $(i \times j)$ $\underline{\mathbf{\Theta}}$ mapping matrix, yields the $(m \times j)$ output matrix $\underline{\mathbf{Y}}$ whose rows comprise $\widehat{\mathbf{y}}^{(m)} \approx \mathbf{y}^{(m)}$. Note that in practice, $\underline{\mathbf{\Theta}}$ consists of a set of matrices. Specifically, for each layer there is a matrix, $\underline{\mathbf{W}}$, of size $(\mathcal{N}_{\ell-1} + 1) \times \mathcal{N}_\ell$, comprising the optimized layer weights augmented with a column containing the biases for each layer. To determine the activation of the neurons in the next layer, its transpose is multiplied by the preceding layer's activation:

$$\underline{\mathbf{a}}^{(\ell)} = \left[\underline{W}^{(\ell)}\right]^{\mathrm{T}} \cdot \underline{\mathbf{a}}^{(\ell-1)} + \underline{\mathbf{b}}^{(\ell)} \tag{7}$$

The loss function was minimized across the training data set using an adaptive moment (Adam) estimation optimization method (Kingma and Ba, 2014), which is a stochastic gradient-based optimizer. SciKit-Learn's default value of $\alpha = 0.0001$ was used; a small α helps minimize bias. The MLP model found weights and biases that minimized the loss function.

3.1.2. Support vector machine model

A One-versus-One (OvO) SVM multi-class classification strategy was applied to replicate the discrete T values from the SWAN simulations (Knerr et al., 1990). SVMs are binary classifiers but the OvO approach can handle multiple classes. OvO assembles all combinations of characteristic-T pairs into $\mathrm{N}(\mathrm{N}-1)/2 = 55$ binary classes (because there were $\mathrm{N} = 11$ discrete values for T in the entire design matrix). The training data were divided into groups corresponding to the 55 combinations of $T_i T_j$ pairs (e.g., all $\underline{\mathbf{x}}$ vectors associated with T_1 and T_2, then all $\underline{\mathbf{x}}$ vectors associated with T_1 and T_3 all the way up to all $\underline{\mathbf{x}}$ vectors associated with $T_{\mathrm{N}-1}$ and T_{N}). A hinge-loss function (including L_2 regularization) was defined as (Moore and DeNero, 2011):

$$\vartheta = \frac{1}{\mathrm{n}} \sum_{k=1}^{\mathrm{n}} \left\{ \max\left[0, 1 - \psi\left(\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}}^{(k)} + b\right)\right] \right\}^2 + \alpha \left\|\underline{\mathbf{w}}\right\|_2^2. \tag{8}$$

where $\psi = \pm 1$ distinguishes members of the $T_i T_j$ pair (i.e., $\psi = +1$ for T_i and $\psi = -1$ for T_j) and n is the number of training-data vectors in the $T_i T_j$ group. Note that the hinge loss is 0 when $\psi(\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}} + b) \geq 1$. The cost function is large when T_i is associated with $\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}} + b \ll 0$ or when T_j is associated with $\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}} + b \gg 0$. Minimization of the cost function results in selection of weights that avoids associating T_j with $\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}} + b \gg 0$ or T_i with $\underline{\mathbf{w}}^{\mathrm{T}} \cdot \underline{\mathbf{x}} + b \ll 0$. A unique $\underline{\mathbf{w}}$ is issued for each of the 55 training-data groups. These optimized $(\underline{\mathbf{w}}, b)$ are aggregated into the $(i \times \mathrm{N})$ mapping matrix, $\underline{\mathbf{\Theta}}$. Next, the dot product of a training-data vector with the mapping matrix yielded 55 rational numbers, which were translated into "votes." When the rational number was positive for a $T_i T_j$ pair, a vote was cast for T_i (for which $\psi = +1$) while a negative rational number was a vote for T_j (for which $\psi = -1$). The T with the most votes was nominated as the characteristics T returned from the machine learning model.

3.1.3. Training data sets

Design matrices were developed by completing 11,078 SWAN model runs dating back to the archived extent of ROMS currents nowcasts (from April 1st, 2013 to June 30th, 2017). A Python script downloaded the ROMS netcdf data and extracted u and v velocities from the nodes in its 3-km grid that were within the SWAN model domain (see the 357 black circles in Fig. 1). Although ROMS ocean-current nowcasts are available every 6 h, these data were linearly interpolated to generate ocean-current fields every 3 h to increase the number of possible SWAN models and expand the training data set. Available NOAA wave-condition data from Buoy 46042 were downloaded at times corresponding to the 3-h increments of ROMS ocean-currents data. Finally, TWC historical wind speeds were downloaded at 0.25° increments throughout the SWAN

model domain (12 turquoise circles in Fig. 1). There were 1,090 occasions when data from Buoy 46042 or ROMS currents were missing and no model was run for those times. A MatLab script was written to develop input files including the primary SWAN input file where wave conditions (H_s, T, and D) on the boundaries were specified along with the spatially variable ROMS ocean-currents files (357 values each for u and v), and the spatially variable TWC winds files (12 values each for easterly and northerly wind components). These data were assembled into $\underline{\mathbf{x}}$ vectors comprising: the three wave-characteristic boundary conditions (H_s, T, and D), 357×2 ocean currents, and 12×2 wind speeds. Overall, the design matrix $\underline{\mathbf{X}}$ had 11,078 rows and 741 columns.

A different $\underline{\mathbf{Y}}$ was required for the MLP and OvO algorithms. For the MLP algorithm, $\underline{\mathbf{Y}}$ was composed of the 11,078 SWAN model runs (rows), each of which comprised 3,104 wave heights (columns) defining the H_s field. For the OvO algorithm, only a $\mathbf{y}$ vector of 11,078 characteristic T values was supplied.

Note that in practice, data in the design matrices were pre-processed. Specifically, $\underline{\mathbf{X}}$ underwent a global normal transform (i.e., all constituent members were scaled so that their overall distribution was Gaussian with zero mean and unit variance). No pre-processing of MLP's $\underline{\mathbf{Y}}$ was required, but the OvO's $\mathbf{y}$ was recast into labels 0 through $\mathrm{N}-1$ corresponding to characteristic T_1 through T_{N} where $\mathrm{N} = 1$.

The $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ data were always randomly shuffled into two groups to form the training data set composed of 90% of the 11,078 rows of data with the test data set the remaining 10%. The mapping matrix $\underline{\mathbf{\Theta}}$ was calculated using the training data set and then applied to the test data set and the RMSE between test data vector, $\mathbf{y}$, and its machine-learning representation, $\widehat{\mathbf{y}}$ was calculated.

For the MLP approach, training was performed many times to identify the number of hidden layers and the number of nodes per layer that yield the lowest overall RMSE. In practice, the MLP model offers two data files; the first describes the normal transform applied to $\underline{\mathbf{x}}$, the dot product of which was taken with the data included in the second file defining $\underline{\Theta}$.

The OvO algorithm need only be supplied with $\underline{\mathbf{X}}$ and the column vector of characteristic T values assembled as $\mathbf{y}$. The data were again split into two groups with 90% of the $\underline{\mathbf{x}}$ vectors randomly assembled into the training data set with the rest reserved for testing. The OvO model returns three files; the first describes the normal transform applied to $\underline{\mathbf{x}}$, the dot product of which was taken with mapping matrix $\underline{\Theta}$ defined in the second file, and the third file defines how to label $\mathbf{y}$ and the same file was used for converting $\widehat{\mathbf{y}}$ back into the characteristic T.

3.1.4. Significant wave heights

MLP regression was used to reproduce the SWAN-generated H_s. Initially, between two and 10 layers were investigated with anywhere from two to 3,000 nodes per layer, but it was quickly determined that fewer nodes (between 10 and 40 per layer) tended to yield smaller RMSEs for the test data used to evaluate each MLP layer/node combination. RMSEs ranged from 18 cm (six layers with 10 nodes each) to 9 cm (three layers with 20 nodes each), which was less than 5% of the average H_s. Although not appropriate for direct comparison, the RMSE for the MLP model was up to 80% lower than those in the SWAN model with respect to the three buoy data sets (see Table 2). Fig. 5 summarizes the performance of the machine learning model at replicating SWAN-predicted wave heights showing the average H_s from each of the 11,078 SWAN model runs and the corresponding machine-learning estimates. A line fit to the data in Fig. 5 has slope 1.002, so bias was negligible. Moreover, note that even for the 14 instances where SWAN-simulated average $H_s > 6$ m, the machine learning representation was quite accurate (RMSE = 14 cm). In fact, the absolute relative error in the machine learning representation of wave height actually decreased with increasing average H_s. That is, although the RMSE tended to increase with average H_s, it did so at a slower rate than H_s itself.

Fig. 6 are two representative heat maps of the differences between
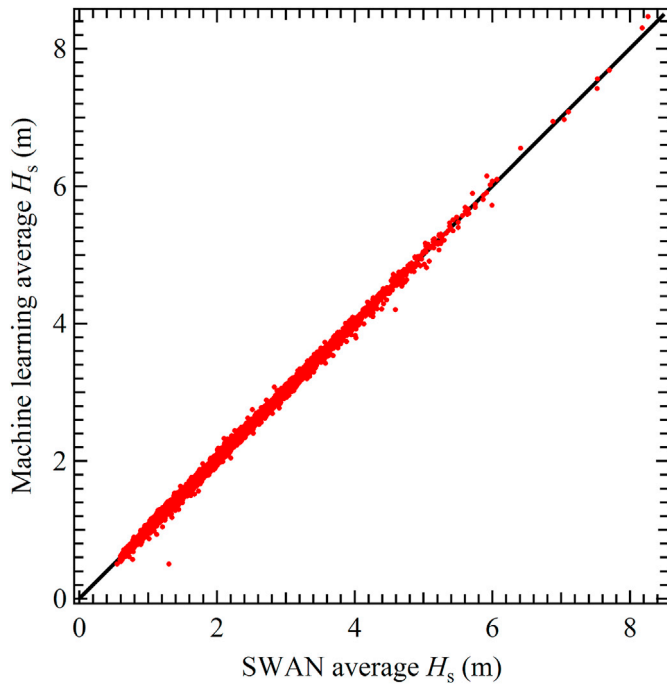
Fig. 5. Cross plot of SWAN and machine learning average H_s for each of the 11,078 model runs. The final MLP architecture comprised three layers of 20 nodes each.

SWAN-simulated H_s fields and the machine-learning equivalents selected from the 11,078 SWAN model runs. In the left snapshot, there remain some local trends where the machine learning model under-predicted H_s by up to 15 cm in the Bay and under-predicted H_s by 15 cm near the south boundary around longitude 237.9° although RMSE = 6 cm. The right snapshot actually had a higher RMSE (14 cm), but does not reveal strong location-based trends. Many of the H_s-differences snapshots were visualized and no clear trend in their patterns could be identified. (Perhaps this would benefit from another application of machine learning model like a convolutional neural network.)

A k-fold cross validation (Bengio and Grandvalet, 2004) was conducted on the MLP model to assess whether overfitting occurred and to ensure that the results can be generalized to an independent data set. In k-fold cross validation, the $\underline{\mathbf{X}}$ input matrix is randomly partitioned into k equal-sized subsamples. Of the k subsamples, a single subsample was retained as the validation data for testing the model, and the remaining $k-1$ subsamples were used as training data. If some of the k-fold RMSEs were notably higher than others, it could indicate over fitting or other model failings (i.e., lucky selection of the test data set). Dividing the data set into 10 shuffled 90%:10%::train:test data sets (10 k-fold iterations) yielded RMSEs ranging from 8.0 to 10.2 cm for the test data set, which were always slightly outperformed, as expected, by the RMSEs for the training data set (ranging from 7.5 to 9.8 cm).

3.1.5. Characteristic wave period

Effectiveness of the OvO model was evaluated according to the percentage of correctly identified characteristic T values. No bias was observed in the OvO results and the percentage of characteristic T accurately identified in the test data set was slightly higher than that from the best combination of layers and nodes in the MLP model (two layers of 10 nodes). OvO correctly identified the characteristic T 90.1% of the time in the test data set. The cross plot of characteristic T from SWAN and the OvO representation shown in Fig. 7 reveals that of the 166 times (out of 11,078 input vectors) that the machine learning model missed the characteristic T, it did so by one discretized T increment except for two instances that were missed by two increments. Running 10 iterations of k-fold testing correctly identified the characteristic T 90% of the time in the test data (and 98.6% in the overall data set with an RMSE below 0.1 s).

It is worth noting that characteristic T values were initially supplied to an MLP model after preprocessing $\widehat{\mathbf{y}}$ into discrete values between 0 and $N-1$. Again, various numbers of layers and nodes were used to replicate characteristic T values and the percentage of accurate results in the test data was assessed. While the correct percentage of test data was comparable to the OvO scheme, there was a bias toward over-prediction; hence this approach was abandoned. Of course, if SWAN did not specifically calculate discrete values for T or if there was significant variability in the T field, then an MLP model would be the more appropriate approach.

4. Discussion

Now that the mapping matrix (or vector) and the pre- and post-processor functions from the machine learning models have been
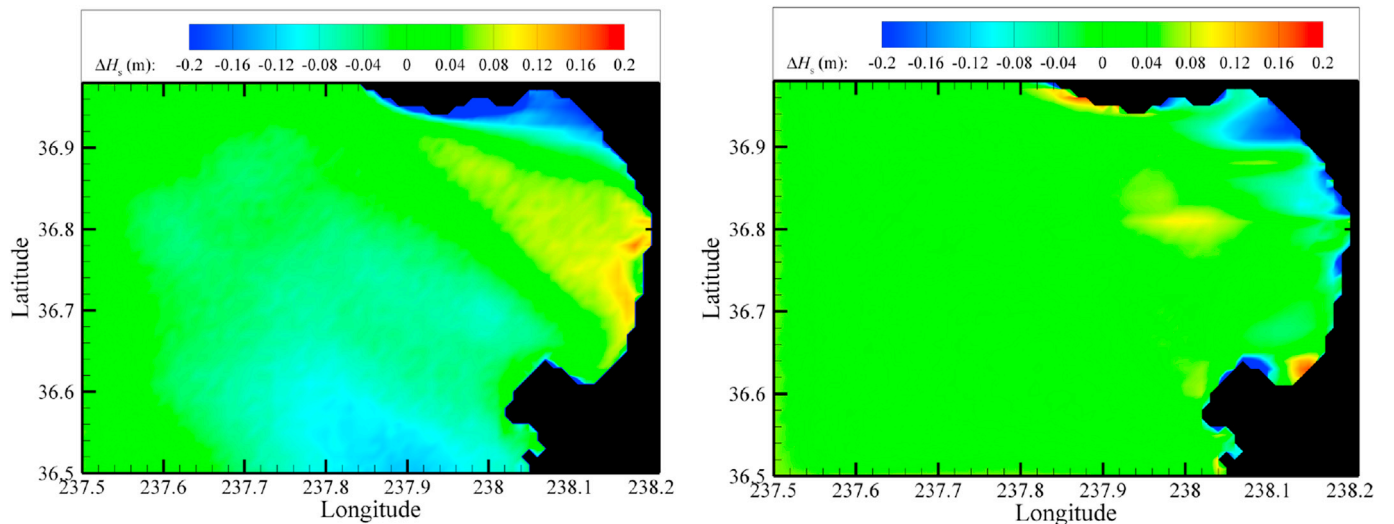

Fig. 6. Two representative heat maps of the differences between SWAN- and machine-learning-simulated H_s selected from the 11,078 SWAN model runs. The wave-height-differences snapshot on the left shows some trends of local discrepancy (in this image, RMSE is 6 cm) not evident in the right figure, which actually has a higher RMSE (14 cm in this image). Nevertheless, most of the domain has near-zero RMSE with local deviations nearer the shoreline where secondary effects are most prominent.
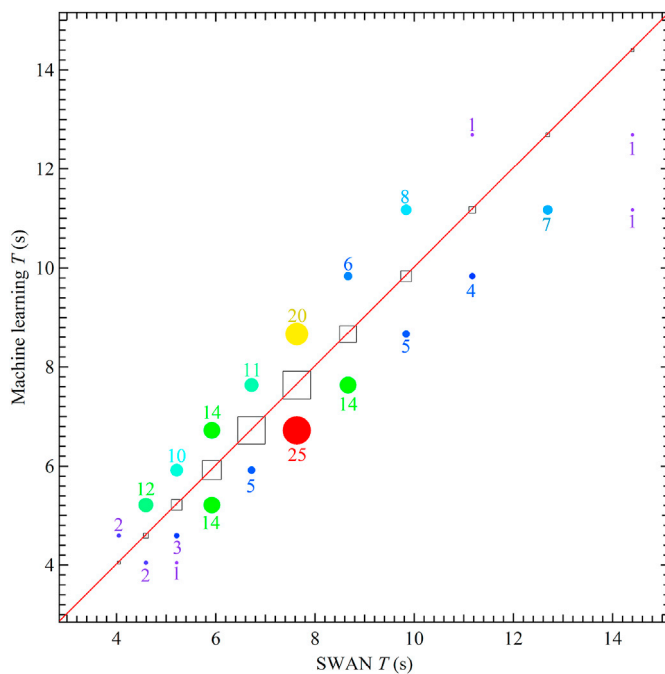
Fig. 7. Cross plot of SWAN and machine learning characteristic *T*. The number of missed *T* values is indicated next to the symbol.

Table 3
Forecast meta-data.

Source	Data	Resolution	Times issued (UTC)	Forecast range (days)
NOAA WAVE WATCH III	H_s, *T*, *D*	0.25°	0, 6, 12, 18	7.5
CeNCOOS ROMS	Ocean currents	3 km	3, 9, 15, 21	2
TWC	E and N winds	1.1 km	Hourly	14

developed, they can act as a surrogate for the SWAN model. Instead of historical data being used to build the input vector, **x**, now forecast data can be used. To run in a forecast mode, the same model inputs used to force the SWAN must be assembled into a vector, which is multiplied by the machine learning mapping matrix to yield the H_s field and characteristic *T*. Such data are part of the Coastal Data Information Program (O'Reilly et al., 2016) in the form of WAVEWATCH III-forecasted wave conditions available for up to the next 10 days. Also, ROMS-simulated ocean-currents and winds forecasts are available for the next 48 h from the CeNCOOS (Integrated Ocean Observing System, 2017) and The Weather Company (2017), respectively. These forecast data are readily available for the Monterey Bay area as summarized in Table 3. In fact, an ensemble-based, machine learning framework is under development by the authors that provides improved wave-condition forecasts for Monterey Bay (O'Donncha et al., 2018).

The machine learning models can be executed to quickly generate the H_s field and corresponding characteristic *T*. Computationally, this only requires $L + 1$ matrix multiplications. In fact, for a 48-h forecast (16 simulations, one every 3 h), SWAN simulations on a single-core processor took 583 s (112 s on eight cores) while the machine-learning equivalent took 0.086 s to calculate the H_s field and 0.034 s to calculate the characteristic *T* (a total of 0.12 s on a single processor) — well over three orders of magnitude (485,833%) faster than the running the full physics-based SWAN models. In fact, the operation requiring the most wall-clock time is actually loading the machine learning matrix files into memory. "Edge computing" (Shi et al., 2016) would have these mapping matrices pre-loaded into memory resulting in nearly instantaneous wave-field forecasts.

It is noted that the machine learning models presented here are specific to the Monterey Bay region and will require re-training to apply to other locations. Of course, running a physics-based model at a new site requires grid generation and assembly of all boundary and forcing conditions along with all of the attendant effort. However, what is important is that the framework necessary to develop this technology has been presented for the first time for wave modeling. It is expected that these sort of data-centric machine learning approaches will grow increasingly common in the near future.

5. Conclusions

Machine learning models were developed as an accurate and computationally efficient surrogate for the SWAN model to predict H_s fields and characteristic *T*. The machine learning MLP nonlinear mapping of 741 inputs to 3,104 H_s outputs required three hidden layers with 20 nodes each; the ReLU activation function was used. An SVM was used to classify characteristic *T*. Using appropriately trained mapping matrices determined from supervised training of machine learning models, surrogates, which are really just matrix multiplication calculations, run over 4,000 times faster than the physics-based SWAN model and yield similarly accurate representations of wave conditions in the domain of interest. Thus, the machine learning models can act as a rapid, efficient wave-condition forecast system. These forecasted wave conditions can be used to estimate surf conditions or the power-generation potential of WECs. Ultimately, it is envisioned that such machine learning models could be installed locally on a WEC thereby facilitating it being its own forecast system. Moreover, the buoy itself can collect wave-condition data that can be used to update the machine learning models. As machine learning technologies improve, they can be adapted to compile a continuous stream of real-time data collected locally with available forecasts into ever-evolving and improving machine learning model parameters. In fact, such procedures are already frequently implemented with "on-line learning" (Cesa-Bianchi et al., 2004).

In this study, the SWAN model was run assuming steady-state conditions. This was appropriate for Monterey Bay as it is dominated by deep-water waves from offshore (Chang et al., 2016). However, other sites subject to more dynamic conditions (e.g., larger domains or high wind energy), require non-steady-state approaches. This increases the complexity of the machine learning algorithm as outputs (wave conditions) are no longer dependent only on inputs (boundary waves, winds and currents) at the current point in time, but also at previous points in time. This requires machine learning approaches with the ability to recognize patterns in time-series data such as recurrent neural networks or long short-term memory (Goodfellow et al., 2016). Future work is focused on investigating approaches to extend this study to non-steady-state conditions.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.coastaleng.2018.03.004.

References

Bengio, Y., Grandvalet, Y., 2004. No unbiased estimator of the variance of k-fold cross-validation. J. Mach. Learn. Res. 5, 1089–1105.

Bidlot, J.-R., Holmes, D.J., Wittmann, P.A., Lalbeharry, R., Chen, H.S., 2002. Intercomparison of the performance of operational ocean wave forecasting systems with buoy data. Weather Forecast. 17, 287–310.

Bieker, H.P., Slupphaug, O., Johansen, T.A., 2007. Real-time production optimization of oil and gas production systems: A technology survey. SPE Prod. Oper. 22, 382–391.

Blanning, R.W., 1975. The construction and implementation of metamodels. Simulation 24, 177–184.

Browne, M., Strauss, D., Castelle, B., Blumenstein, M., Tomlinson, R., Lane, C., 2006. Empirical estimation of nearshore waves from a global deep-water wave model. Geosci. Rem. Sens. Lett. IEEE 3, 462–466.

Browne, M., Castelle, B., Strauss, D., Tomlinson, R., Blumenstein, M., Lane, C., 2007. Near-shore swell estimation from a global wind-wave model: Spectral process, linear, and artificial neural network models. Coast Eng. 54, 445–460.

Cahill, B., Lewis, T., 2014. Wave period ratios and the calculation of wave power. In: Proceedings of the 2nd Marine Energy Technology Symposium, pp. 1–10.

Cesa-Bianchi, N., Conconi, A., Gentile, C., 2004. On the generalization ability of on-line learning algorithms. IEEE Trans. Inf. Theor. 50, 2050–2057.

Chang, G., Ruehl, K., Jones, C., Roberts, J.D., Chartrand, C.C., 2016. Numerical modeling of the effects of wave energy converter characteristics on nearshore wave conditions. Renew. Energy 89, 636–648.

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.

Dushaw, B.D., Egbert, G.D., Worcester, P.F., Cornuelle, B.D., Howe, B.M., Metzger, K., 1997. A TOPEX/POSEIDON global tidal model (TPXO.2) and barotropic tidal currents determined from long-range acoustic transmissions. Prog. Oceanogr. 40, 337–367.

EPRI, 2011. Mapping and Assessment of the United States Ocean Wave Energy Resource. Technical Report 1024637, Palo Alto, CA.

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, 1175–1181.

FAO, 2016. The State of the World Fisheries and Aquaculture 2016. Contributing to the Food Security and Nutrition for All. Technical Report. Food and Agriculture Organization of the United Nations.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Haas, K., Ahn, S., Neary, V.S., Bredin, S., 2017. Development of a wave energy resource classification system. Waterpower Week 1–5.

Hodur, R.M., 1997. The Naval Research Laboratory's coupled ocean/atmosphere mesoscale prediction system (COAMPS). Mon. Weather Rev. 125, 1414–1430.

Integrated Ocean Observing System, 2017. Central and Northern California Ocean Observing System. CeNCOOS.

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

Kleijnen, J.P., 2009. Kriging metamodeling in simulation: A review. Eur. J. Oper. Res. 192, 707–716.

Knerr, S., Personnaz, L., Dreyfus, G., 1990. Single-layer learning revisited: A stepwise procedure for building and training a neural network. Neurocomp. NATO ASI Ser. (Ser. F: Comp. Syst. Sci.) 68, 41–50.

Komen, G.J., Cavaleri, L., Donelan, M., 1996. Dynamics and Modelling of Ocean Waves. Cambridge University Press.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

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.

Makarynskyy, O., 2004. Improving wave predictions with artificial neural networks. Ocean Eng. 31, 709–724.

Mallet, V., Stoltz, G., Mauricette, B., 2009. Ozone ensemble forecast with machine learning algorithms. J. Geophys. Res.: Atmos. 114.

Mei, C.C., Stiassnie, M., Yue, D.K.-P., 1989. Theory and Applications of Ocean Surface Waves: Part 1: Linear Aspects. Part 2: Nonlinear Aspects. World Scientific.

Moore, R., DeNero, J., 2011. L_1 and L_2 regularization for multiclass hinge loss models. In: Symposium on Machine Learning in Speech and Language Processing, pp. 1–5.

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

Ocean Energy Systems, 2016. Annual Report: Ocean Energy Systems 2016. Technical Report 1024637.

O'Donncha, F., Zhang, Y., James, S.C., 2018. An integrated framework that combines machine learning and numerical models to improve wave-condition forecasts. J. Mar. Syst.

O'Hagan, A., 2006. Bayesian analysis of computer code outputs: A tutorial. Reliab. Eng. Syst. Saf. 91, 1290–1300.

O'Reilly, W., Olfe, C.B., Thomas, J., Seymour, R., Guza, R., 2016. The California coastal wave monitoring and prediction system. Coast Eng. 116, 118–132.

Patterson, J., Thomas, J., Rosenfeld, L., Newton, J., Hazard, L., Scianna, J., Kudela, R., Mayorga, E., Cohen, C., Cook, M., et al., 2012. Addressing Ocean and Coastal Issues at the West Coast Scale through Regional Ocean Observing System Collaboration. MTS/IEEE OCEANS'12, pp. 1–8.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. SciKit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Peres, D., 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.

Razavi, S., Tolson, B.A., Burn, D.H., 2012. Review of surrogate modeling in water resources. Water Resour. Res. 48. W07401.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Network. 61, 85–117.

Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L., 2016. Edge computing: vision and challenges. IEEE InternetThings J. 3, 637–646.

Song, Y., Haidvogel, D., 1994. A semi-implicit ocean circulation model using a generalized topography-following coordinate system. J. Comput. Phys. 115, 228–244.

The SWAN Team, 2006. SWAN Scientific and Technical Documentation. Technical Report SWAN Cycle III version 40.51. Delft University of Technology.

The Weather Company, 2017. The Weather Company.

Tolman, H.L., et al., 2009. User manual and system documentation of WAVEWATCH III™ version 3.14, Technical note. MMAB Contrib. 276, 220.

US DOE, 2016. Energy Department Announces Investment in Wave Energy Test Facility.