KSOE

Review Article

Journal of Ocean Engineering and Technology 36(3), 194-210, June, 2022
https://doi.org/10.26748/KSOE.2022.007

pISSN 1225-0767
eISSN 2287-6715

Review on Applications of Machine Learning in Coastal and Ocean Engineering

Taeyoon Kim[1] and Woo-Dong Lee[2]

[1]*Research Professor, Institute of Marine Industry, Gyeongsang National University, Tongyeong, Korea*
[2]*Associate Professor, Department of Ocean Civil Engineering, Gyeongsang National University, Tongyeong, Korea*

KEY WORDS: Machine learning, Data-driven model, Coastal engineering, Prediction, Sensitivity analysis

ABSTRACT: *Recently, an analysis method using machine learning for solving problems in coastal and ocean engineering has been highlighted. Machine learning models are effective modeling tools for predicting specific parameters by learning complex relationships based on a specified dataset. In coastal and ocean engineering, various studies have been conducted to predict dependent variables such as wave parameters, tides, storm surges, design parameters, and shoreline fluctuations. Herein, we introduce and describe the application trend of machine learning models in coastal and ocean engineering. Based on the results of various studies, machine learning models are an effective alternative to approaches involving data requirements, time-consuming fluid dynamics, and numerical models. In addition, machine learning can be successfully applied for solving various problems in coastal and ocean engineering. However, to achieve accurate predictions, model development should be conducted in addition to data preprocessing and cost calculation. Furthermore, applicability to various systems and quantifiable evaluations of uncertainty should be considered.*

1. Introduction

Humans have attempted to understand various phenomena that occur in the sea many years ago. However, these phenomena are difficult to elucidate even after several centuries because of complex interactions involving physical, chemical, and biological processes (Dawarakish et al., 2013). Hence, various methods have been formulated through statistical analysis, spectrum analysis, time series analysis, empirical formulas based on mathematical model experiments, and other mathematical and physical analyses. However, the derivation of accurate results is limited owing to the complex interrelationships among numerous parameters in nature (Goldstein et al., 2019).

As we enter the era of the Fourth Industrial Revolution, machine learning (ML) models that identify and predict statistical structures from input and output data using computers to solve numerous engineering problems in the natural world are garnering significant attention. ML, a field of artificial intelligence (AI), is an inductive method that identifies rules through learning using data and results, instead of using a conventional program method that derives results from rules and data. ML techniques can easily solve complex engineering problems and enable the regression analysis of nonlinear relationships. ML demonstrates clear advantages over other conventional regression methods because it adopts a specific algorithm that can learn from the input data and provides accurate results via the output (Salehi and Burgueño, 2018). ML-based predictive models include various algorithms such as neural networks, decision trees, support vector machines (SVM), and gradient boosting (GBR). In coastal engineering, studies based on ML-based algorithms are increasingly conducted to predict wave formation, wave breaking, tidal changes, hydraulic properties around structures, and changes in beach profiles (Deo and Jagdale, 2003; Panizzo and Briganti, 2007; Kankal and Yuksek, 2012).

This paper introduces ML models and reviews various studies that predict significant parameters in coastal engineering such as waves, wave breaking, hydraulic properties around structures, and beach profile changes. This paper focuses on regression analysis studies that involve continuous variables for parameters during the supervised learning of ML models, whereas studies pertaining to classification involving categorical variables are omitted. Furthermore, the basic concepts and basic contents of various ML models are introduced, and the technological trends and application examples of ML models in the coastal engineering field are described.

Received 30 March 2022, revised 26 April 2022, accepted 26 May 2022
Corresponding author Woo-Dong Lee: +82-55-772-9126, wdlee@gnu.ac.kr

2. Machine Learning Model

As we enter the era of the Fourth Industrial Revolution, interest in AI and ML is increasing. ML, which is a field of AI, trains computers human thinking and cognition methods such that the computers can perform recognition and inference on their own without preset judgment criteria for all variables. General ML algorithms include supervised learning, unsupervised learning, and reinforcement learning (Fig. 1). Two supervised learning models exist: regression and classification. Data classification and prediction are determined based on the characteristics of both the input and dependent variables. Representative supervised learning algorithms include artificial neural network (ANN), SVM, and random forest (RF). Unsupervised learning is a method of predicting results for new data by clustering patterns or features from unlabeled data. It is primarily used for clustering and dimensionality reduction, e.g., k-means clustering and principal component analysis. This paper focuses on the application cases of coastal and marine engineering for supervised learning among ML algorithms.

2.1 Linear Regression (LR) Model

The LR model uses linear parameters and offers easy and quick analyses. The LR model was developed more than a century ago and has been widely used over the past few decades. However, it yields low accuracy results for data that exhibit nonlinear relationships. LR generates a regression model using one or more features and obtains parameters w and b that minimize the mean squared error (MSE) between the experimental value (y) and predicted value ($\hat{y}$) (Eqs. (1)–(2)).

$$\hat{y} = w[0] \times x[0] + w[1] \times x[1] + \ldots + w[p] \times x[p] + b \tag{1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

2.1.1 Lasso regression

In some cases, the conventional LR method results in overfitting, i.e., the predictive performance is unsatisfactory when new data are provided. Hence, lasso regression was developed, which limits models by force using the L1 regulation, as follows (Eq. (3)):

$$E = MSE + penalty = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\sum_{j=1}^{m}|w_j| \tag{3}$$

where m is the number of weights, and α is the $penalty$ parameter. This equation obtains w and b that minimize the sum of the MSE and $penalty$.

2.1.2 Ridge regression

Ridge regression is a model in which the L2 regulation term is added to solve the overfitting problem of the LR model. This model not only fits the data of the learning algorithm, but also ensures that the weights of the model are minimized (Eq. (4)).

$$E = MSE + penalty = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\frac{1}{2}\sum_{j=1}^{m}|w_j|^2 \tag{4}$$

The difference between lasso regression and ridge regression is that the weights are zero in lasso, whereas in ridge, the weights are approximately zero but not exactly zero. Hence, the lasso regression offers high accuracy only if some of the input variables are important, whereas the accuracy of the ridge model will be high if the importance of the input variables is similar in general.

2.2 ANN

An ANN is an information processing structure in the form of a network modeled based on the human nervous system, where simple functional processors are interconnected on a large scale. The

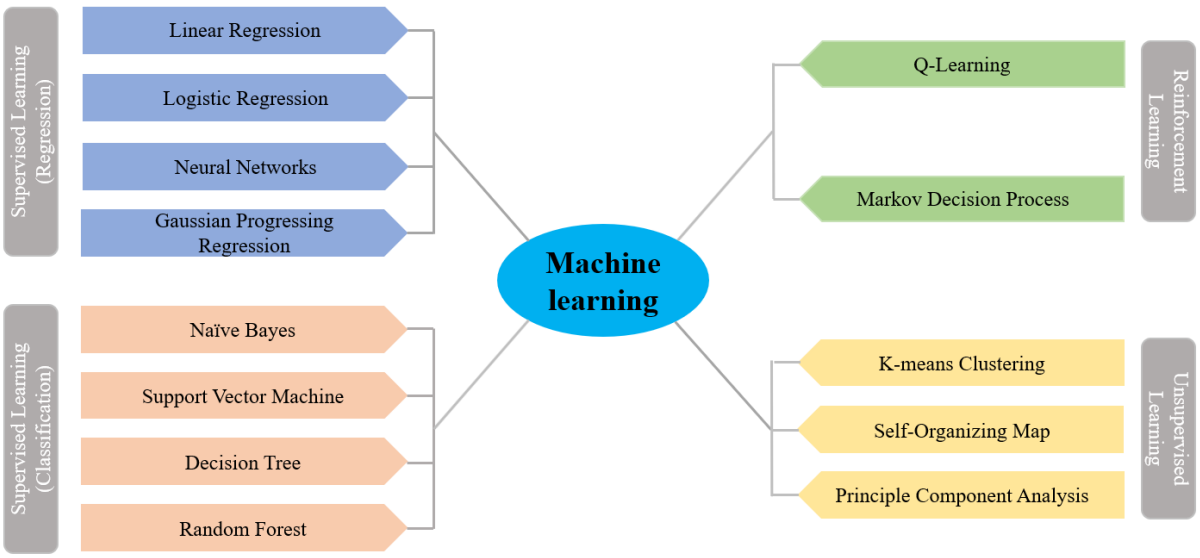


Fig. 1 Machine learning algorithms (Liu et al., 2021)

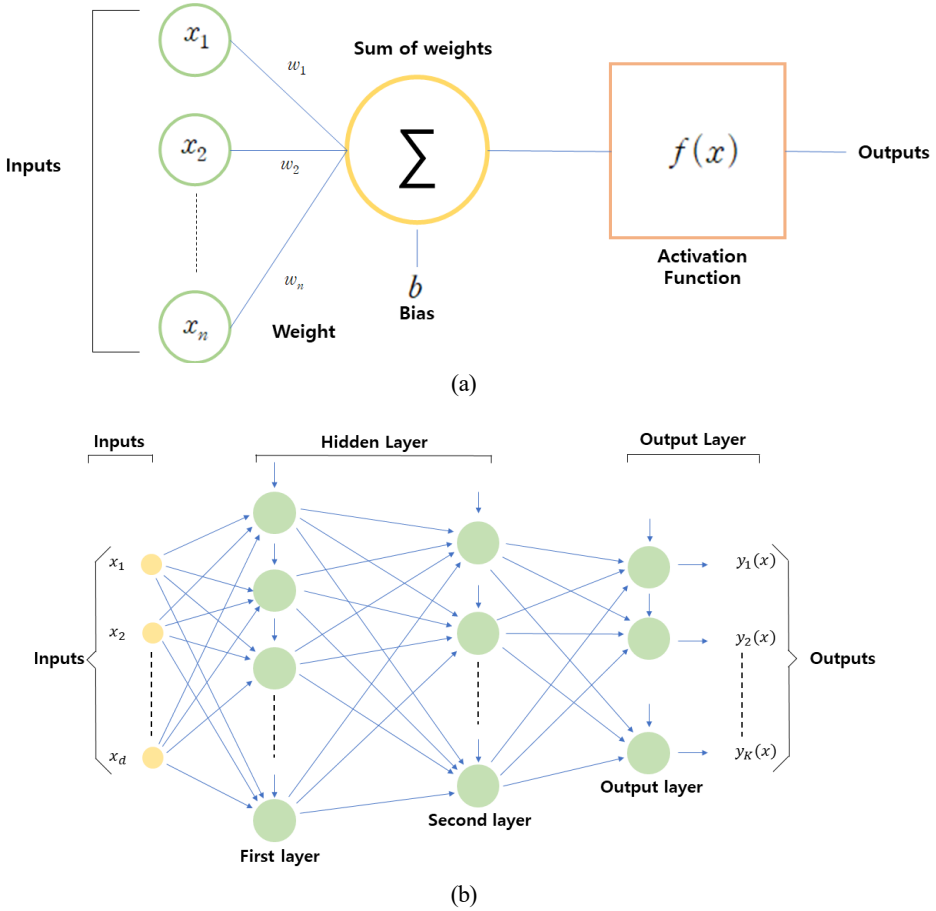


Fig. 2 Artificial neural network: (a) Neural network diagram of element; (b) typical layout of neural network

perceptron, which is the basis of deep learning, is a structure designed to deliver information based on a threshold by issuing a weighting signal for input values through the imitation of neuron behaviors in brain cells. Fig. 2 shows the structure of a neural network. A neural network comprises an input layer, a hidden layer, and an output layer, and its nodes are interconnected by weights. In an ANN, the values are transmitted from the input layer to all nodes of the hidden layer in a feedforward method. Moreover, the output values of all nodes of the hidden layer are transmitted to all nodes via the coupling and activation functions. Learning proceeds by redistributing weights between neurons through the backpropagation algorithm such that they converge in a direction in which the errors are reduced to minimize the difference between the predicted and experimental values.

In the data processing process of the neural network, each node multiplies the input value by a weight and then passes the output value through the activation function to the next node to output the result, as expressed in Eq. (5).

$$N_j = \sum_{i=1}^{N_i} w_{ij} x_i + \theta_j \tag{5}$$

where w is the weight, θ the bias, and x the input value.

The activation function renders the neural network nonlinear and enables a nonlinear analysis of the result calculated from the coupling function via the calculation of the activation function, which is a nonlinear function. Furthermore, the activation function is key for adjusting the gradient during activation training. Various activation functions are used in neural networks, including the linear, sigmoid, tanh, exponential, softmax, rectified linear unit (ReLU), ELU (Exponential Linear Unit), and SELU(Scaled Exponential Linear Unit) functions.

2.3 SVM

The SVM was introduced by Boser et al. (1992), who were inspired by the concept of statistical learning theory. SVM regression performs training to include the maximum amount of data within the specified margin error limit line. The limit line adjusts the width of the margin based on the hyperparameter. Here, the margin implies the distance between the grain boundary and support vector. The procedure of applying the SVM to the regression problem is as follows (Eqs. (6)–(7)). First, the dataset for training is distinguished.

$$\{(x_1, y_1), \ldots (x_n, y_n)\},\ x \in R_n,\ y \in r \tag{6}$$

where x is the input variable, y the output variable, R_n the n-dimensional vector space, and r the one-dimensional vector space.

The loss function of $\varepsilon^{\in sensitive}$ can be expressed as follows:

$$L_{\varepsilon}(y)=0 \text{ For } |f(x)-y|<\varepsilon \text{ or } L_{\varepsilon}(y)=|f(x)-y|-\varepsilon \tag{7}$$

Based on the equation above, if the predicted value is within the expected range, then the loss function is 0; if the predicted value is outside the expected range, then the loss function of $\varepsilon^{\in sensitive}$ is defined such that the loss is equal to the absolute value of the standard deviation minus ε. The main purpose of the vector machine is to provide the deviation of ε in the actual output value and to obtain a uniform function $f(x)$.

Finally, $f(x)$ can be expressed as follows (Eqs. (8)–(9)):

$$f(x)=\sum_{i=1}^{nsv}(\alpha_i-\alpha_i^*)K(x_i,x_j)+b \tag{8}$$

$$K(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right) \tag{9}$$

where α_i and α_i^* are Lagrangian coefficients, and $K(x_i,x_j)$ represents the kernel function. In general, homogeneous polynomials, polynomial kernels, Gaussian radial basis functions, and hyperbolic tangent functions are used as the kernel function.

2.4 Gaussian Process Regression (GPR)

The Gaussian process regression (GPR) model is a probability model based on nonparameteric kernels. The Gaussian process $f(x)$ is a set of random variables $f(x)$ in the range of $\{(x_i,y_i);\, x_i\in R^d\}$, where a finite number of randomly selected variables, $f(x_1),\ldots,f(x_m)$, among them exhibits a combined Gaussian density (Na et al., 2017). The GPR model for the new input vector ($x_{\neq w}$) and training data predicts $y_{\neq w}$. The linear regression model is expressed as follows:

$$y=x^{\top}\beta+\varepsilon \tag{10}$$

where $\varepsilon\sim N(0,\sigma^2)$, and the error variance (σ^2) and coefficient (β) are calculated based on data. The GPR model describes predictions by introducing potential variables in the Gaussian process $f(x_i)$, $i=1,2,\ldots,n$ and the explicit basic function $h(x)$. The covariance function of the hidden variable provides flexibility to the response, and the basic function transmits the input of x to the p-dimensional feature space. The Gaussian process is a random variable set that comprises a finite number of Gauss distributions. If $\{f(x),\ x_i\in R^d\}$ is a Gaussian process and n contains the observed values $x_1,x_2,\ldots,x_n$, then the random variables $f(x_1),f(x_2),\ldots,f(x_n)$ exhibits a Gaussian distribution. The Gaussian process is defined by the mean function ($m(x)$) and covariance function ($k(x,x')$).

$$m(x)=E(f(x)) \tag{11}$$

$$k(x,x')=E[(f(x)-m(x))(f(x')-m(x'))] \tag{12}$$

$$f(x)\sim GP(m(x),k(x,x')) \tag{13}$$

$$h(x)^{\top}\beta+F(x) \tag{14}$$

Here, $h(x)$ is a basic function set that transmits the existing feature vector x of R^d to the new feature vector $h(x)$.

$$P(y_i|f(x_i),x_i)\sim N(y_i|h(x_i)^{\top}\beta+f(x_i),\sigma^2) \tag{15}$$

Therefore, the GPR model can be represented as a probabilistic model as in Eq. (15), and the hidden variable $f(x_i)$ is introduced to observe each x_i (Koo et al., 2016).

2.5 Ensemble Method

The ensemble method is developed to improve the performance of the classification and regression tree. It generates an accurate prediction model by creating several classifiers and combining their predictions. In other words, it is a method of deriving a high-accuracy prediction model by combining several weak classifier models, instead of using a single strong model.

The ensemble models can be primarily categorized into bagging and boosting models. Bagging is a method of reducing variance using the averaging or voting method on the results predicted using various models, whereas boosting is a method of creating strong classifiers by combining weak classifiers.

2.5.1 RF

RF is a method of improving the large variance of the decision tree and the large performance fluctuation range. It combines the concept and properties of bagging with randomized node optimization to overcome the disadvantages of existing decision trees and improves generalization. The process of extracting bootstrap samples and generating a decision tree for each bootstrap sample is similar to bagging. However, it is different from the conventional decision tree in that a method of randomly extracting predictors and creating an optimal split within the extracted variables is used instead of selecting the optimal split within all predictors for each node (Kim et al., 2020). In other words, RF combines the randomization of predictors while determining slightly different training data through bootstrap to obtain maximum randomness. Hence, several low-importance learners are created. The important hyperparameters used in an RF include max_features, bootstrap, and n_estimator. The max_features parameter refers to the maximum number of features to be used in each node. The bootstrap allows redundancy in data sampling conditions for each classification model. The n_estimator refers to the number of trees to be created in the model (Kim and Kim, 2020).

2.5.2 Boosting method

Boosting is a technique for creating a strong classifier from a few

weak classifiers. It is a model created by boosting weights on data at the boundary. The adaptive boosting (AdaBoost) algorithm is the most typically and widely used algorithm among ensemble learning methods. Specifically, it is one of the boosting series in ensemble learning. In AdaBoost, after a weak classifier is generated using the initial training data, the distribution of the training data is adjusted based on the prediction performance afforded by the training of the weak classifier. The weight of the training sample with low prediction accuracy is increased using the information received from the classifier in the previous stage. In other words, the training accuracy is improved by adaptively changing the weights of samples with low prediction accuracy in the previous classifier. Finally, a strong classifier with slightly better performance is created by combining these weak classifiers with low prediction performance. GBR is a method of sequentially adding multiple models such as the AdaBoost model. The most significant difference between the two algorithms is the method by which they recognize weak classifiers. AdaBoost recognizes values that are more difficult to classify by weighting them. By contrast, GBR uses a loss function to classify errors. In other words, the loss function is an index that can evaluate the performance of the model in learning specific data, and the model result can be interpreted based on the loss function used.

AdaBoost can be used for both classification and regression. In general, when a regression problem is considered, the training data set can be expressed as follows:

$$\phi = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_m, Y_m)\} \tag{16}$$

where $(X_i, Y_i)(i = 1, \ldots, m)$ is the ith sample of the training dataset, m the total number of samples, X_i the input data vector value, and Y_i the output data value.

Next, we can train a weak classifier $G(X)$ using a specific learning algorithm, and the relative prediction error e_i of each sample is expressed as follows:

$$e_i = L(Y_i, G(X_i)) \tag{17}$$

where $L(\bullet)$ is the loss function. In general, three options are available: linear loss, square loss, and exponential loss. For a simple explanation, the linear loss is applied as follows:

$$e_i = \frac{|Y_i - G(X_i)|}{E} \tag{18}$$

where $E = \max|Y_i - G(X_i)|$ is the maximum absolute error of all samples.

The performance of one weak classifier will inevitably be unsatisfactory. Hence, the objective of AdaBoost is to sequentially generate weak classifiers $G_k(X), k = 1, 2, \ldots, N$, and then combine them. A strong classifier $H(X)$ is composed of a few combination strategies. For regression analysis, the combination is expressed as follows:

$$H(X) = \nu \sum_{k=1}^{N} \left(\ln \frac{1}{a_k}\right) g(X) \tag{19}$$

where a_k is the weight of the weak classifier ($G_k(X)$), $g(X)$ is the median of all $a_k G_k(X)$, k = 1,2,...,N, and $\nu \in [0,1]$ is used as a regulatory factor or to prevent overfitting. Both the weak classifier ($G_k(X)$) and weight a_k are generated using the modified value of the existing learning data. As such, the distribution weight of each sample is adjusted based on the error predicted by the previous weak classifier ($G_{k-1}(X)$). Incorrectly predicted samples are repeatedly trained by increasing the weight such that they will be prioritized in the next learning process. During the iteration of k = 1,2,...,N, the weak classifier ($G_k(X)$) and relative prediction error e_{ki} are calculated using Eq. (19). Subsequently, the total error rate e_k is expressed as follows (Eq. (20)):

$$e_k = \sum_{i=1}^{m} e_{ki} \tag{20}$$

Furthermore, the weight a_k of the weak classifier can be represented as follows (Eq. (21)):

$$\alpha_k = \frac{e_k}{1 - e_k} \tag{21}$$

Finally, for the next step learning, the weight distribution of each sample ($w_{k+1,i}$) is updated again as follows (Eq. (22)):

$$w_{k+1,i} = \frac{w_{k,i} \alpha_k^{1-e_{ki}}}{\sum_{i=1}^{m} w_{k,i} \alpha_k^{1-e_{ki}}} \tag{22}$$

Between the two types of weights (w_k, α_k) defined above, the first (w_k) involves training the data sample and is used to enable better training in the next step after the weight of the incorrectly predicted sample is increased. The second (α_k) implies a weak classifier and is used to such that a more accurate weak classification will impose a greater effect on the final result. AdaBoost provides a stronger framework than specific learning algorithms because it does not provide a specific form of the weak classifier $G(X)$. Theoretically, every type of ML regression algorithm can be used as a weak classifier in AdaBoost.

3. Application of ML in Coastal and Marine Engineering Field

3.1 Wave Prediction

Accurate wave estimations can be applied to coastal engineering, marine transportation, and leisure sports. For example, the transportation route can be optimized by reducing the transportation

time through accurate wave information prediction, which can provide accurate prediction information regarding the generation of wave energy. Furthermore, useful information can be provided to surfers in a surf zone by providing wave information at a coast. Significant wave height is an important parameter in coastal port structure design and construction. Most physics-based models are applied to estimate such wave information. However, wave estimation studies based on ML models have increased recently (Deo and Naidu, 1999; Balas et al., 2004; Mahjoobi et al., 2008; Shahabi et al., 2016; Oh and Suh, 2018; Garcia et al., 2021).

James et al. (2018) developed an ML model to predict the characteristics of wave distributions. Data were generated via a few thousand rounds of iterative learning using a physics-based model known as the simulating waves nearshore (SWAN) model. Furthermore, a model for predicting the significant wave height and peak period using a multilayer perceptron and an SVM was proposed. A total of 741 input variables were applied considering the wave conditions at the interface (H_s, T_s, D), flow distribution within the grid (u,v), wind speed, and wind direction. Meanwhile, 11,078 data points were used for training, where two output variables (i.e., the significant wave height and peak period) calculated using the SWAN model were applied (Table A1). The result shows that the ML model reproduced more than 90% of the wave characteristics of the physics-based model, with an MSE of 9 cm. Moreover, the computation time is shorter compared with that afforded by the SWAN model; therefore, it is expected to be a promising alternative to the physics-based model. Fig. 3 shows a heat map presenting the difference ($\triangle H$) between the value predicted by the representative ML model and the value calculated using the SWAN model for the result of the ML model based on data derived from the calculations of 11,078 cases using the SWAN model. The image on the left shows the result of underestimating the significant wave height to a maximum of 15 cm near the bay, although the RMSE is 6 cm. However, the image on the right indicates an RMSE of 14 cm, although the error near the bay is smaller, and a clear location-based trend is not shown. Although statistical values such as the RMSE are important, the reliability of the model may differ by the application purpose. Therefore, the accuracy should be further improved through additional data analysis.

Shamshirband et al. (2020) constructed three ML models, i.e., the ANN, support vector regression (SVR), and the extreme learning machine (ELM) for wave height estimation and compared their performances with the results of the SWAN model. The input variable applied to the ML models was the near-surface wind speed, and the output variable was the significant wave height measured at the Bushehr and Assulayeh Ports of the Persian Bay. These variables were applied to training. The prediction performance of the ELM model was excellent, and the ML-based model of the Bushehr Port was reliable. However, for the Assaluyeh Port, the prediction accuracy for the significant wave height was low, and a correction was performed to improve the efficiency. Furthermore, the result of the ML model underestimated the extreme values. Hence, accurate input values and data preprocessing technology are necessitated to improve the result. The results of the SWAN model underestimated the extreme wave height. Both models require improvement for predicting extreme wave conditions. The ML-based model can be implemented at a low computational cost without requiring the depth information. However, unlike the SWAN model, it requires a separate model to predict the wave height of locations other than the two points used for training.

Chen et al. (2021) proposed a new surrogate model developed using the RF method, which is an ML model, based on spatial wave data estimated using the SWAN numerical model. Twelve input variables were used for model training: the significant wave height (H_s), mean

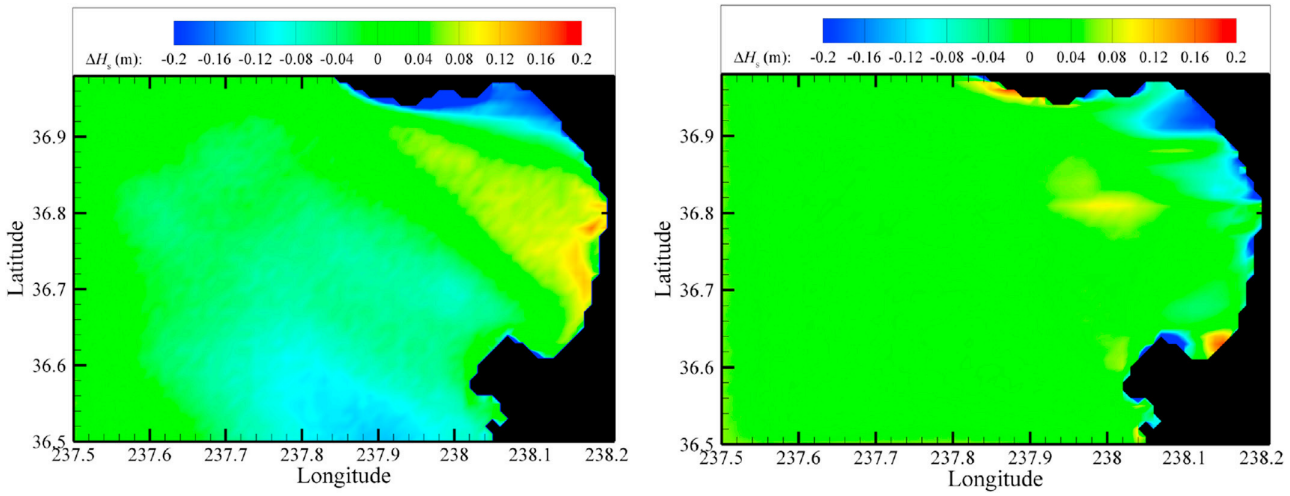


Fig. 3 Figure from James et al. (2018), who used ANN and SVM to predict the significant wave height (H_s) and wave period (T). Differences between SWAN- and machine-learning-simulated H_s selected from 11,078 SWAN model simulations are shown. Image of wave height differences on the left shows local discrepancy trends (RMSE = 6 cm in this image), which are not evident in the figure on the right, which in fact has a higher RMSE (i.e., 14 cm). Nevertheless, the domain shows primarily near-zero RMSEs, with local deviations at locations closer to the shoreline, where secondary effects are the most prominent.

wave direction (m_{Dir}), period (T_z), and peak period (T_p) at three buoys. For the output variable, the spatial wave information derived via SWAN calculation was used. The result of the ML model agreed better with the on-site buoy observation values than with the result of the SWAN model, and the computation time of the ML model was 100 times shorter than that of the SWAN model.

Kim et al. (2010) calculated the expected damage of an inclined breakwater using an ANN. The ANN, which uses the tide level and deep-sea waves as input, was trained to predict shallow-sea significant waves. They proved that a high degree of expected damage can be estimated within a short duration by calculating the shallow sea waves.

Kang and Oh (2019) investigated the prediction of swell wave generation using the RF, logistic regression, the K-nearest neighbor algorithm, the ANN, and the SVM. Changes in the water temperature, atmospheric pressure, and tide level were confirmed as primary variables for predicting swell high waves. Furthermore, the RF model performed the best (prediction accuracy: 88.6%).

Park et al. (2020) estimated the significant wave height of an X-band radar using an ANN; this method was demonstrated to be superior over the conventional wave height observation method. The result of a comparative analysis based on Hujeong Beach in Uljin confirmed the high accuracy of the calculated significant wave height.

Lee et al. (2020) conducted a wave breaking prediction study using an open-source ML algorithm to quantitatively predict wave breaking on a coast. The prediction results for the wave breaking wave height and depth by their trained neural network showed better prediction performance compared with the results calculated using the conventional empirical formula.

3.2 Tide Level Prediction

Accurate predictions of the tide level are crucial because the tide level significantly affects navigation, leisure activities, and coastal ecosystems. Tides refer to the periodic ascent and descent of the Earth's sea level due to tidal forces caused by the sun and moon. Thus, the tide level is an important parameter in terms of coastal engineering, maritime safety, and maritime activities. Various other factors such as the wind speed and atmospheric pressure must be considered in addition to the tide level. A harmonic analysis method in which many sine wave components are superimposed is generally used to predict the tide level; however, the effects of time-dependent factors are difficult to consider in this method. This paper introduces research cases that apply an ML model for tide level prediction in coastal and marine engineering.

The conventional harmonic decomposition method requires a significant amount of observational tidal data. Moreover, the parameters of the harmonic analysis model are estimated using the least-squares method based on data obtained for a long duration (i.e., more than 1 month). Lee (2004) constructed an ANN model using short-term measurements for tide level prediction and applied the $\cos(w_n t_j)$ and $\sin(w_n t_j)$ functions for 69 tidal components as input variables. Consequently, the primary components were determined based on two months' worth of measurement data. A comparison between the ANN and harmonic analysis models showed improved accuracy by the ANN model. Moreover, when 15 d of observation data were applied to training, the model presented prediction results that were applicable for predicting the tide levels for 1 year.

These tidal changes involve complex processes that are affected by not only the movement of celestial bodies, but also by nonperiodic meteorological factors such as wind, atmospheric pressure, and water temperature. However, the effect of time-dependent factors cannot be considered using the conventional harmonic analysis method. Therefore, Li et al. (2018) developed a tide level prediction model

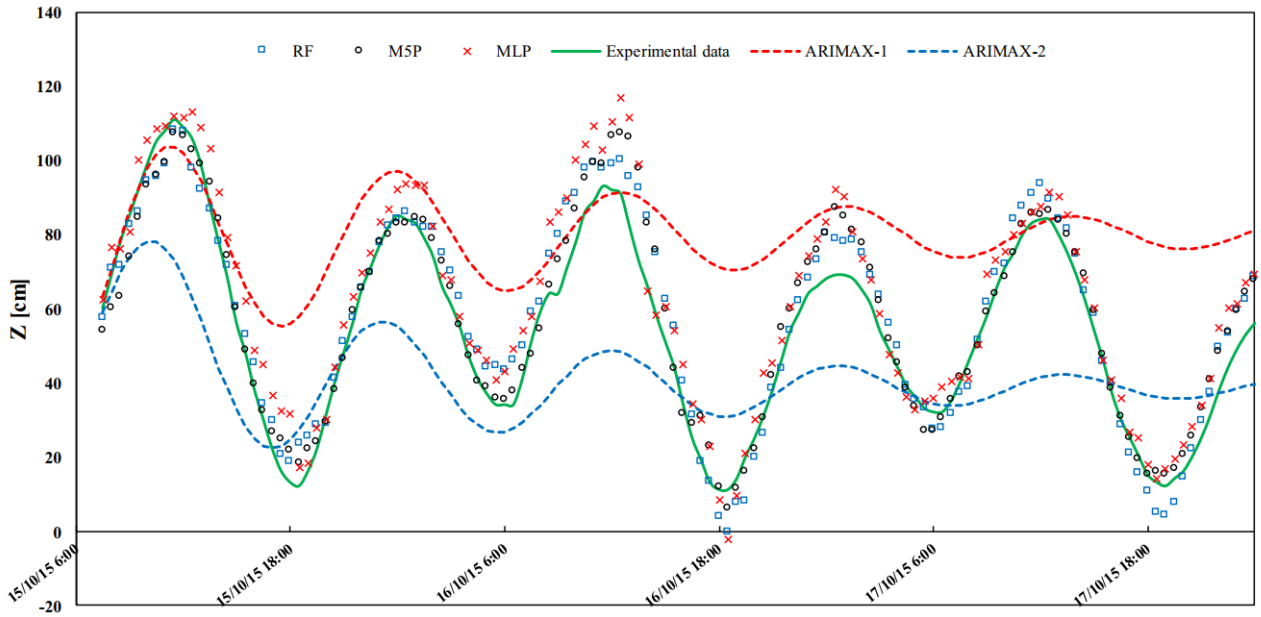


Fig. 4 Figure from Granta and Nunno (2021), who used M5P, RF and ANN to predict tide level. Forecast of tide fluctuations with 5 h advance: comparison between ML-based and ARIMAX models.

using the ELM to consider various nonlinear factors such as wind, air pressure, and water temperature. They presented results with higher accuracy and time efficiency than those yielded by conventional harmonic analysis models.

Granta and Nunno (2021) suggested a tide level prediction model using M5P, RF, and ANN algorithms (which are ML models). A total of 28 input variables were used in that study, including the astronomical tide (AT), wind speed (WS), barometric pressure (BP), and previously observed tide levels (Z_{-24} to Z_{-1}) to construct a tide level prediction model. Fig. 4 shows the tide level prediction results using the three ML models and the ARIMAX regression analysis. The M5P model showed a high coefficient of determination of 0.924–0.996. The result of sensitivity analysis showed high-level prediction results without considering meteorological factors (WS, BP), including for exceptionally high water levels. These results suggest that the training dataset can be continuously updated and applied to sea level fluctuations caused by climate change and subsidence.

3.3 Estimation of Design Variables

Several phenomena pertaining to wave–structure interactions exist, such as reflections in the structure, wave breaking on the slope and at the front, dissipation, wave runups and rundowns, transmitted waves, and overtopping. Therefore, techniques for understanding and quantitatively estimating these phenomena are required for the design of structures. Various studies have been conducted to estimate the primary design variables (Goyal et al., 2014; Lee and Suh, 2019; Lee and Suh, 2020; Etemad-Shahidi et al., 2016; Najafzadeh et al., 2014). Herein, we provide examples of applying the ML model for calculating the stability number, overtopping rate, wave transmission coefficient, and reflection coefficient.

3.3.1 Estimating stability number

Kim and Park (2005) constructed a stability number calculation model for rubble-mound breakwater using an ANN. For the input variables of the model, seven parameters were applied, including porosity (P), the number of wave attacks (N_W), damage level (S_d), structure slope ($\cos\alpha$), wave height (H_s), wave period (T_s), dimensionless water depth (h/H_s), and spectral shape (SS). Moreover, the stability number (N_s) was set as the output model. The result shows that the ML model demonstrated higher accuracy in predicting the stability number and damage level compared with the results obtained using the conventional empirical formula. Therefore, it can be utilized for design purposes.

Yagci et al. (2005) modeled the damage rates of different breakwaters using ANN, multiple LR, and fuzzy models. The experimental results yielded by the multiple LR model were unsatisfactory. However, they reported that the neural network and fuzzy model results can be estimated through interpolation for missing values.

Etemad-Shahidi and Bonakdar (2009) constructed a stability prediction model for rubble-mound breakwater using the M'5 model. Five input parameters were applied to the model: porosity (P), the number of wave attacks (N_W), damage level (S_d), surf similarity coefficient (ξ_m), and dimensionless water depth (h/H_s). Based on comparison, the results obtained showed higher accuracy compared with those obtained using the conventional Van der Meer empirical equation. A new equation was derived based on the M5' model, which proved to be useful for engineering design.

Based on the experimental data of Van der Meer et al. (1988), Koc et al. (2016) suggested a stability number prediction model for breakwater using the genetic algorithm. The experiment result showed that the genetic algorithm afforded better prediction performance than the empirical equation for the stability number.

3.3.2 Estimating overtopping rate

EurOtop is a representative result of a study that predicted the overtopping rate using an ML tool. In the Crest Level Assessment of Coastal structures by Full-scale Monitoring, Neural Network Prediction, and Hazard Analysis on Permissible Wave Overtopping (CLASH) project (De Rouck et al., 2004), an ANN model was developed to predict the mean overtopping rate, q (Pullen, 2007).

Van Gent et al. (2007) constructed an ANN-based prediction model to estimate the overtopping rates of various coastal structures. A database comprising approximately 10,000 mathematical model data points obtained from the European CLASH project was used for model training. A complexity factor and a reliable factor (RF) were introduced to increase data reliability. Data with low reliability or high complexity were excluded from training. Subsequently, the remaining data were converted to $H_{m0,toe}$ = 1 m by applying Froude's law of similarity to match with the mathematical model test results. Fig. 5 describes the parameters used for training. As input variables, 15 parameters describing wave characteristics (e.g., the significant wave height, average period, and wave direction) and factors pertaining to the structural shape (e.g., ridge depth, crest width, and slope) were applied. The mean overtopping rate (q) was set as a dependent variable. The result suggests that the ANN model is sufficiently applicable for modeling the correlation between the input variables related to overtopping and the average overtopping rate in coastal structures. However, all datasets were applied for training without segmenting the dataset in this study, and data with $q < 10^{-6}$ m^3/s/m were excluded from training. Therefore, the generalization of the model is likely to be difficult.

Subsequently, errors in the CLASH database were corrected, and more than 17,000 datasets were expanded through the Innovative Technologies for Safer European Coasts in a Changing Climate project (Zanuttigh et al., 2014). The calculation for the overtopping rate and the estimated results for uncertainty were presented using an ANN model. In previous studies, data with $q < 10^{-6}$ m^3/s/m were removed as measurement errors by experiment were assumed to have increased. However, Zanuttigh et al. (2016) categorized all data into three quantitative classifiers and constructed a training and a prediction model. The result showed improved prediction accuracy compared

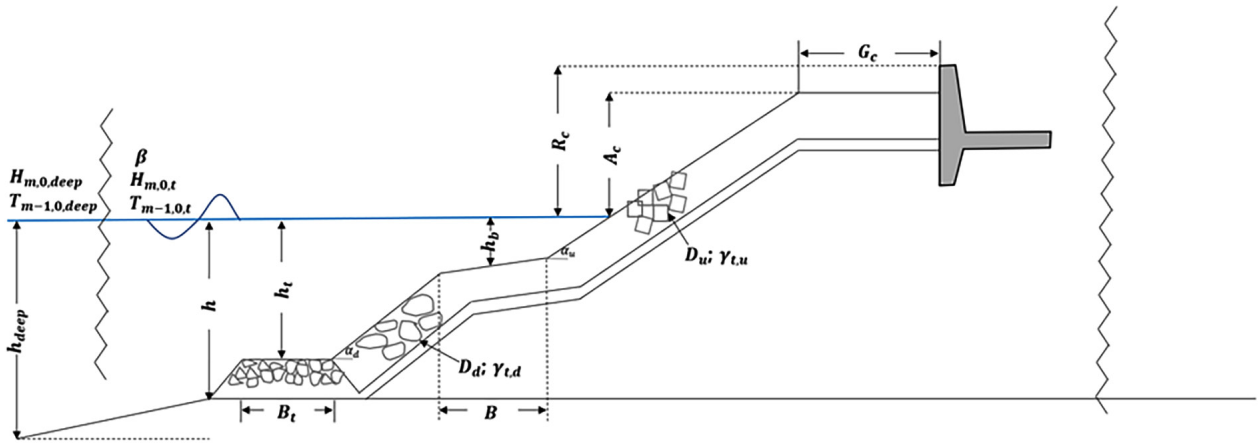


Fig. 5 Schematic illustration of structure based on CLASH, including geometrical and hydraulic parameters

with the results of previous studies, and the generalization performance was high even for data not used for training.

Den Bieman et al. (2020) and Den Bieman et al. (2021) constructed an overtopping rate prediction model using extreme gradient boosting (XGBoost)—an ML model. The result shows that the prediction error was 2.8 times lower than that of the existing neural network model. Moreover, they conducted variable importance analysis via feature engineering. The result shows that the XGBoost model can be successfully applied as an alternative to the ANN model.

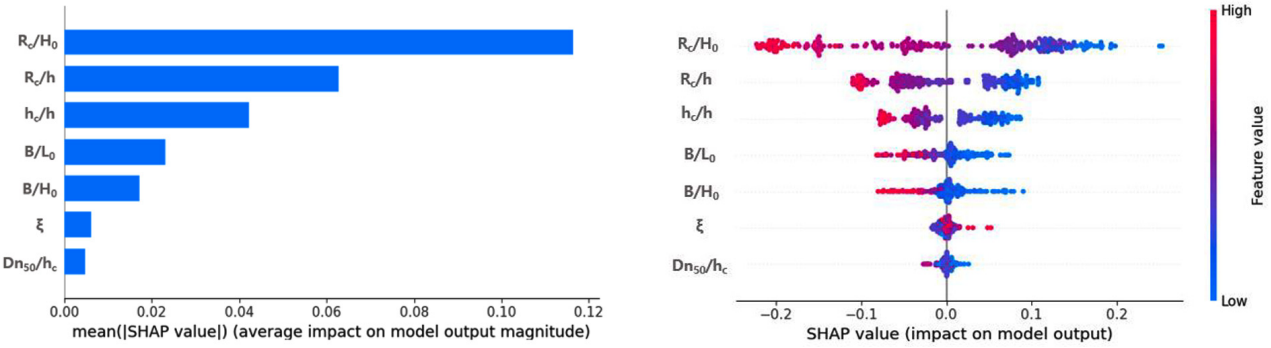
Hosseinzadeh et al. (2021) constructed a mean overtopping rate prediction model for an inclined breakwater using GPR and SVR models, which are two kernel-based ML models. The result showed that the accuracy of the GPR model was higher than that of the conventional ANN model and empirical formula. Furthermore, they derived an optimal combination of input variables through sensitivity analysis and demonstrated that the prediction yielded is more accurate than that afforded by the combination of input variables by Van der Meer et al. (2018).

3.3.3 Wave trasmission and reflection coefficients

Formentin et al. (2017) proposed a prediction model for the mean overtopping rate and wave transmission/reflection coefficients (K_t and K_r) using the CLASH database to predict wave–structure interactions. An ANN was applied (as an ML model), and 15 nondimensional input variables were applied while considering the structural characteristics (geometric structure, amplitude, and roughness) and wave attack (wave slope and wave direction). Moreover, the overtopping rate, wave transmission coefficient, and reflection coefficient were set as output variables. The result showed that the ANN model afforded a higher prediction accuracy than the existing empirical formula and can be useful for design.

Kuntoji et al. (2018) proposed a prediction model for the wave transmission rate of underwater breakwaters using SVM and ANN models. They developed a model by applying eight input variables, including the wave slope (H_i/gT^2) and relative reef width. The prediction result showed that the SVM model to which the kernel function was applied afforded a higher accuracy than the ANN model with an coefficient of dtermination (R^2) value of 0.984.

Gandomi et al. (2020) estimated the wave transmission and reflection coefficients of permeable breakwater structures using a genetic algorithm, an ANN, and an SVM. Seven input variables including the porosity, relative wave height, and wave slope were



(a) **(b)**

Fig. 6 Figure from Kim et al. (2021), who used ML model to predict the wave transmission coefficient. Graphs show the variable importance, where the x-axis represents the average absolute Shapley values of the input variables throughout the data. (a) SHAP feature importance; (b) summary plot (feature effects).

applied as input variables. Moreover, the wave transmission and reflection coefficients were set as output variables. The result showed that the exponential GPR model performed the best for a correlation analysis between experimental and predicted values. They proposed a formula for calculating wave transmission and reflection coefficients using a Gaussian model.

Kim et al. (2021) constructed an ML model for estimating the wave transmission coefficient of low-crested structures using data from a mathematical model used in an experiment conducted through the DELOS project. They adjusted the hyperparameters via grid search for 10 ML models such as GBR, AdaBoost, and Gaussian regression, and selected an ML model suitable for the data. Seven nondimensional input variables such as the relative ridge depth and relative crest width were applied as input variables, whereas the wave transmission coefficient was set as an output variable. In addition, they analyzed the correlation between the input variable and the dependent variable using an explanatory AI technique and determined the dominant factor affecting the prediction of the wave transmission coefficient. Fig. 6 shows the variable importance results for each input variable based on the dependent variable analyzed using the ML analysis tool. The factor related to the ridge depth contributed the most toward the prediction of the output variable. This suggests that the reliability of the model should be improved through model analysis instead of constructing a simple ML model.

3.4 Prediction of Morphological Changes

Studies for further understanding and predicting shoreline fluctuations have been actively conducted, owing to the possibility of increasing coastal erosion acceleration promoted by climate change over the past few decades. The quantitative prediction of coastal erosion and restoration is effective for mitigating erosion risks and is essential for establishing a strategic beach management plan. Therefore, the prediction of beach profile deformation due to waves and beach currents is one of the most important tasks in coastal engineering. Various factors such as wind and waves, beach slope, tide level, sediment particle size, and storm surge frequency can affect beach deformation. Various approaches are available for predicting changes in the beach profile. Sediment movement, erosion, and deposition along a coast are primarily estimated based on empirical formulas; however, the corresponding physical mechanism has not been fully clarified. Recently, data-based ML models have been introduced and used for predicting shoreline fluctuations, barrier islands, and sand dune erosion (Yoon et al., 2013; Wilson et al., 2015; Passarella et al., 2018).

Hashemi et al. (2010) predicted the seasonal variation characteristics of beach profiles using an ANN model based on seven years' worth of beach profile data from 19 stations near Tremadoc Bay. Nine parameters were applied as input variables for the model, i.e., the minimum wind speed, wind direction, continuous storm frequency, storm frequency, significant wave height, significant period, wave, beach slope, and wind duration. In addition, they constructed a model using 12 output variables, i.e., the elevation of 10 points of each cross-sectional profile, the area under each profile curve, and the length of the profile. The result showed that the MSE converged to 0.0007 compared with the observed value, demonstrating the high prediction accuracy of the model in estimating beach variability characteristics. These study results suggest that the ML model can be a more effective tool for predicting changes in the beach profile than the mathematical model for the same points, owing to the complexity and uncertainty associated with the physical understanding of morphological dynamics at the shore. However, the ML model is applicable to only previous measurement data; it cannot be applied easily to abnormal climates or structures that are not reflected in the training data. Therefore, a study combining mathematical models and ML is necessary.

Rigos et al. (2016) constructed a model comprising an ANN using the feedforward method to predict the beach circulation pattern of a coast comprising a reef. The beachrock reef in front of the beach increased the complexity of wave actions and nonlinearities. Legendre polynomials were applied as an activation function to reflect this nonlinearity. Data for training were obtained from long-term time-series data for 10 months from January to November 2014 on the target coast. Six independent variables were applied as input variables, i.e., the ridge depth, structure slopes, structure width, significant wave height, and peak wave period. Subsequently, the model was built by setting the offshore distance as a dependent variable.

López et al. (2017) predicted the sandbar generated on a coast by applying an ANN model. Seven input variables were applied, including the wave characteristics, sediment characteristics, and time data. A model was constructed to predict the location of the sandbar crossing the shore based on six dependent variables at the barrier islands' feature points (start, ridge, and end points). The results showed that the error of the neural network model was lower than that of the general empirical formula for predicting the characteristics of barrier islands.

Montaño et al. (2020) conducted a workshop and contest related to the shoreline fluctuation model "Shoreshop," where participants from 15 international organizations tested and improved the performance of the model for predicting shoreline changes. They presented the result of a modeling contest, in which 19 models including the conventional numerical model were tested using the data pertaining to the daily average shoreline position and beach rotation for approximately 18 years (between 1999 and 2017) in the target sea, Tairua Beach. The result showed that the performance of the ML model was comparable to those of conventional numerical models. In fact, the multiyear variability at shorelines, which is difficult to simulate using conventional numerical models, can be analyzed easily using their model. Fig. 7 shows the results of shoreline fluctuations predicted using a numerical model, an ML model, and a hybrid model. Shoreline fluctuations during extreme events that occurred on a short time scale (~monthly) was difficult to reproduce using the general numerical model. Meanwhile, the ML model adequately reproduced shoreline

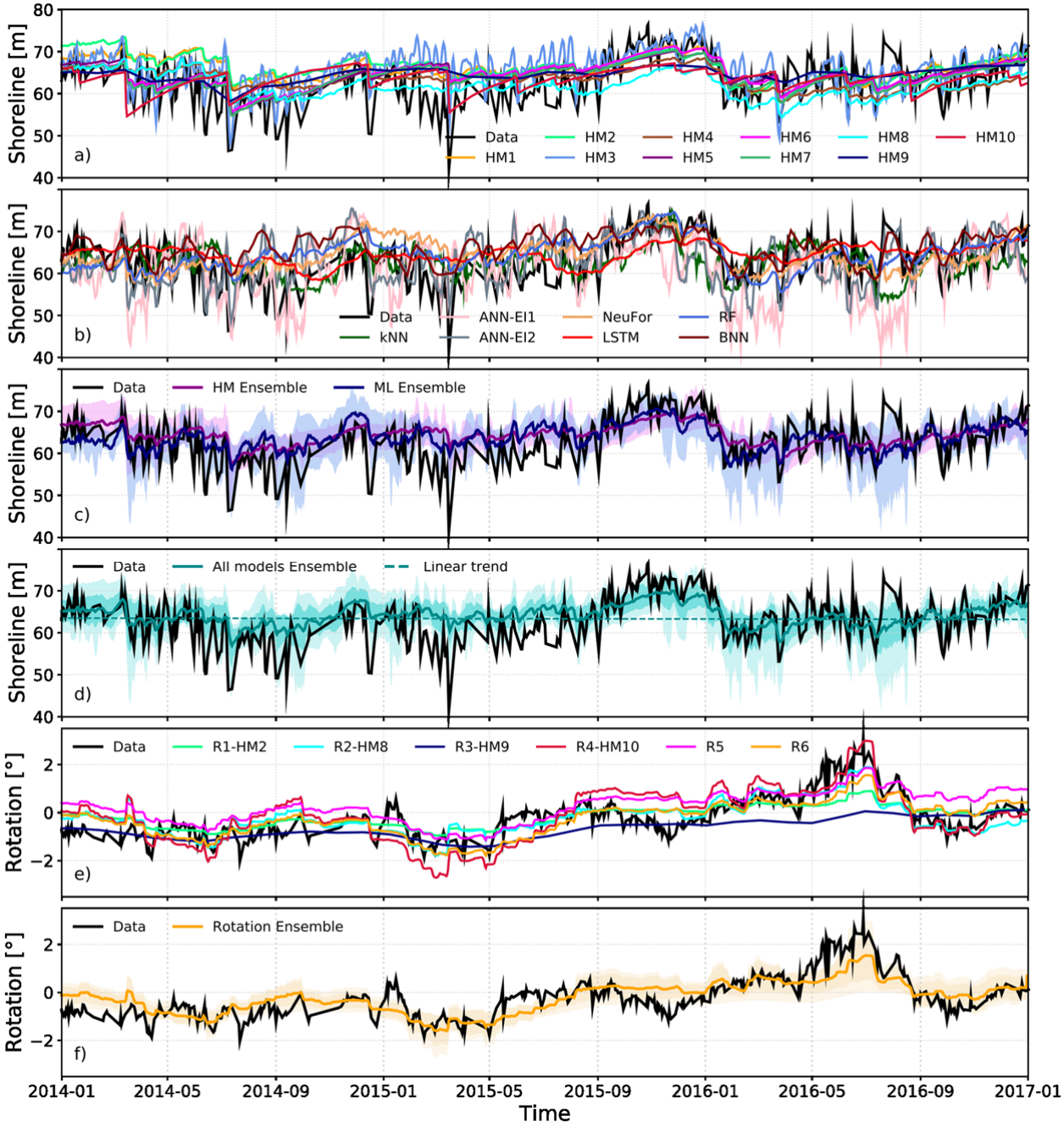


Fig. 7 Figure from Montaño et al. (2020), who used ML and numerical model to predict shoreline evolution. Model outputs (see legends) compared with observations (black): (a) Hybrid models; (b) ML models; (c) HM and ML ensemble; (d) multimodel ensemble; (e) rotation models; (f) hybrid model ensemble for beach rotation. Dark shadows in ensemble figures represent one standard deviation of model prediction. Light shadows represent maxima/minima envelope of model predictions. See Methods section and Supporting Information for model details.

fluctuations in extreme events. However, the ability of the ML model in predicting the shoreline position deteriorated for the case involving data not used for training (2014–2017). Therefore, the ML (inductive) and numerical (deductive) models complement each other in estimating shoreline fluctuations owing to their different approaches. Consequently, the ensemble approach combining the ML and numerical models improved the prediction reliability and reduced the uncertainty of the model.

4. Conclusions

In this study, we examined waves, tide level and sea level fluctuations, design variable estimation, and morphological changes in several studies that applied ML in coastal engineering. Based on extensive studies, the ML model proved to be a reliable in solving problems related to coastal engineering. The ML model can be constructed by learning the correlations between the input and output

variables without basic mathematical and physical understanding pertaining to extremely complex interactions and processes associated with coastal engineering. However, several factors should be considered from the researcher's perspective to implement such highly accurate models, including the following:

(1) Amount of data

A significant amount of training data with various ranges is required to construct an ML model. However, the exact amount of data required to derive meaningful prediction results remains elusive. Goldstein et al. (2019) reported that the performance deteriorates when a significant amount of data is applied to a low-complexity model in certain cases. These results cannot be generalized to all cases. However, the amount of data required for optimal prediction using the empirical knowledge of researchers and the method for managing noise in the data must be analyzed quantitatively.

(2) Data preprocessing

Actual data were acquired from various sources and processes; hence, incomplete data, noise, and inconsistent data that reduce the quality of the dataset may be included in the acquired data. Researchers must perform appropriate data preprocessing to improve data quality to achieve high-performance models. During data preprocessing, the model data should be converted into data suitable for the model through data cleaning, which replaces missing values or removes noise data and outliers. Furthermore, data normalization should be performed to reduce dimension and noise by via feature scale matching. For example, Zanuttigh et al. (2016) introduced a weight factor to the preprocessing process for more than 17,000 data points obtained from the CLASH database and reduced the data impurity via bootstrap sampling. Furthermore, the range of the dependent variables was $10^{-9} < q < 10$ but might be underestimated due to error calculation. Thus, a study was performed to improve the accuracy of the model by converting the dependent variable into $\log(q)$. The reliability and accuracy of the model could be further improved through appropriate data preprocessing based on the results of previous studies.

(3) Model analysis

An ML model is a black box model with a complex structure. Therefore, its intuitive interpretation ability is low for supporting the prediction results. To improve the limitations of such black box models, explainable artificial intelligence (XAI) is currently being conducted. The XAI methodology provides interpretation such that humans can understand the results predicted by ML algorithms. The model is reliable if the basis on which the model derived the prediction results through XAI can be determined. Furthermore, one can determine whether the model has been appropriately trained or whether the data used for training are appropriate. Kim et al. (2021) analyzed a wave transmission coefficient prediction model for low-crested structures using the Shapley additive explanation (SHAP). Simple ML models should be constructed and reliable model analyses should be conducted based on previous results.

(4) Model validation and generalization

ML models generally segregate the data into training and test datasets randomly. However, data under extreme conditions or data with important information may be excluded from the training process, and this possibility should be considered when segregating the data. The reliability of the model should be improved through cross-validation, such as the k-fold cross validation and leave one out cross-validation. Furthermore, the model constructed through verification should perform a generalization process using new data that have not been used for training. Montaño et al. (2020) presented the results of blinding tests on a numerical model and an ML model through workshops and competitions pertaining to "Shoreshop," which is a shoreline fluctuation model. The exchange and dissemination of knowledge among researchers worldwide should be promoted, and problems in coastal engineering should be solved from various angles through such modeling contests.

Coastal engineering researchers can obtain new knowledge and insights regarding data analysis via ML models. However, the ML model is an inductive approach rather than a deductive approach, which is the conventional approach used in numerical models. Hence, it is difficult to generalize the model based on the range and characteristics of the data. For example, in regard to the prediction of morphological changes, generalizing all regions based on a single model is difficult because the data characteristics of a specific region are reflected in the model. Therefore, various problems in the coastal engineering field should be solved using an ensemble model that combines the conventional numerical model with an ML model, and by deriving results that improve prediction performance through a mutual complement of their strengths and weaknesses.

Conflict of Interest

Woo-Dong Lee serves as an editor of the Journal of Ocean Engineering and Technology but does not decide the publication of this article. No potential conflicts of interest relevant to this article are reported.

Funding

This study was conducted under the support of the Korea Research Foundation with funding from the government in 2022 (Ministry of Science and ICT) (No. NRF-2022R1C1C2004838).

References

Balas, C.E., Koc, L., & Balas, L. (2004). Predictions of Missing Wave Data by Recurrent Neuronets. Journal of Waterway, Port, Coastal, and Ocean Engineering, 130(5), 256–265. https://doi.org/10.1061/(ASCE)0733-950X(2004)130:5(256)

Boser, B.E., Guyon, I., & Vapnik, V.N. (1992). A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 5, 144−152. Pittsburgh, ACM. https://doi.org/10.1145/130385.130401

Chen, J., Pillai, A.C., Johanning, L., & Ashton, I. (2021). Using Machine Learning to Derive Spatial Wave Data: A Case Study for a Marine Energy Site. Environmental Modelling & Software, 142, 105066. https://doi.org/10.1016/j.envsoft.2021.105066

De Rouck, J., Van de Walle, B., & Geeraerts, J. (2004). Crest Level Assessment of Coastal Structures by Full Scale Monitoring, Neural Network Prediction and Hazard Analysis on Permissible Wave Overtoppingg - (CLASH). Proceedings of the EurOCEAN 2004 (European Conference on Marine Science & Ocean Technology), Galway, Ireland, EVK3-CT-2001-00058, 261−262.

Dwarakish, G.S., Rakshith, S., & Natesan, U. (2013). Review on Applications of Neural Network in Coastal Engineering. Artificial Intelligent Systems and Machine Learning, 5(7), 324−331.

Den Bieman, J.P., Wilms, J.M., Van den Boogaard, H.F.P., & Van Gent, M.R.A. (2020). Prediction of Mean Wave Overtopping Discharge Using Gradient Boosting Decision Trees. Water, 12(6), 1703. https://doi.org/10.3390/w12061703

Den Bieman, J.P., Van Gent, M.R.A., & Van den Boogaard, H.F.P. (2021). Wave Overtopping Predictions Using an Advanced Machine Learning Technique. Coastal Engineering, 166, 103830. https://doi.org/10.1016/j.coastaleng.2020.103830

Deo, M.C., & Naidu, C.S. (1999). Real Time Wave Forecasting Using Neural Networks. Ocean Engineering, 26(3), 191−203. https://doi.org/10.1016/S0029-8018(97)10025-7

Deo, M.C., & Jagdale, S.S. (2003). Prediction of Breaking Waves with Neural Networks. Ocean Engineering, 30(9), 1163−1178. https://doi.org/10.1016/S0029-8018(02)00086-0

Etemad-Shahidi, A., & Bonakdar, L. (2009). Design of rubble-mound breakwaters using M5 machine learning method. Applied Ocean Research, 31(3), 197−201. https://doi.org/10.1016/j.apor.2009.08.003

Etemad-Shahidi, A., Shaeri, S., & Jafari, E. (2016). Prediction of wave overtopping at vertical structures. Coastal Engineering 109, 42−52. https://doi.org/10.1016/j.coastaleng.2015.12.001

Pullen, T., Allsop, N.W.H., Bruce, T., Kortenhaus, A., Schüttrumpf, H., & van der Meer, J.W. (2007). European Manual for the Assessment of Wave Overtopping. Pullen, T., Allsop, N.W.H., Bruce, T., Kortenhaus, A., Schüttrumpf, H. & van der Meer, J.W.(Eds.), HR Wallingford.

Van der Meer, J.W., Allsop, N.W.H., Bruce, T., De Rouck, J., Kortenhaus, A., Pullen, T., ... Zanuttigh, B. (2018). Manual on Wave Overtopping of Sea Defences and Related Structures: An Overtopping Manual Largely Based on European Research, but for Worldwide Application (2nd ed.). EurOtop. Retrieved from http://www.overtopping-manual.com/assets/downloads/EurOtop_II_2018_Final_version.pdf

Formentin, S.M., Zanuttigh, B., & van der Meer, J.W. (2017). A Neural Network Tool for Predicting Wave Reflection, Overtopping and Transmission. Coastal Engineering Journal, 59(1), 1750006-1-1750006-31. https://doi.org/10.1142/S0578563417500061

Gandomi, M., Moharram, D.P., Iman, V., & Mohammad, R.N. (2020). Permeable Breakwaters Performance Modeling: A Comparative Study of Machine Learning Techniques. Remote Sensing, 12(11), 1856. https://doi.org/10.3390/rs12111856

Goyal, R., Singh, K., & Hegde, A.V. (2014). Quarter Circular Breakwater: Prediction 3 of Transmission Using Multiple Regression 4 and Artificial Neural Network. Marine Technology Society Journal, 48(1).

Goldstein, E.B., Coco, G., & Plant, N.G. (2019). A Review of Machine Learning Applications to Coastal Sediment Transport and Morphodynamics. Earth-Science Reviews, 194, 97−108. https://doi.org/10.1016/j.earscirev.2019.04.022

Gracia, S., Olivito, J., Resano, J., Martin-del-Brio, B., Alfonso, M., & Alvarez, E. (2021). Improving Accuracy on Wave Height Estimation Through Machine Learning Techniques. Ocean Engineering, 236, 108699. https://doi.org/10.1016/j.oceaneng.2021.108699

Granta, F., & Nunno, F.D. (2021). Artificial Intelligence Models for Prediction of the Tide Level in Venice. Stochastic Environmental Research and Risk Assessment, 35, 2537−2548. https://doi.org/10.1007/s00477-021-02018-9

Hashemi, M.R., Ghadampour, Z., & Neill, S.P., (2010). Using an Artifi cial Neural Network to Model Seasonal Changes in Beach Profi les. Ocean Engineering, 37(14−15), 1345−1356. https://doi.org/10.1016/j.oceaneng.2010.07.004

Hosseinzadeh, S., Etemad-Shahidi, A., & Koosheh, A. (2021). Prediction of Mean Wave Overtopping at Simple Sloped Breakwaters Using Kernel-based Methods. Journal of Hydroinformatics, 23(5), 1030−1049. https://doi.org/10.2166/hydro.2021.046

James, S.C., Zhang, Y., & O'Donncha, F. (2018). A Machine Learning Framework to Forecast Wave Conditions. Coastal Engineering, 137, 1−10. https://doi.org/10.1016/j.coastaleng.2018.03.004

Kang, D.H., & Oh, S.J. (2019). A Study of Machine Learning Model for Prediction of Swelling Waves Occurrence on East Sea. Journal of Korean Institute of Information Technology, 17(9), 11−17. https://doi.org/10.14801/jkiit.2019.17.9.11

Kankal, M., & Yuksek, O. (2012). Artificial Neural Network Approach for Assessing Harbor Tranquility: The Case of Trabzon Yacht Harbor, Turkey. Applied Ocean Research, 38, 23−31. https://doi.org/10.1016/j.apor.2012.05.009

Kim, D.H., & Park, W.S. (2005). Neural Network for Design and Reliability Analysis of Rubble Mound Breakwaters. Ocean Engineering, 32, 1332−1349. https://doi.org/10.1016/j.oceaneng.2004.11.008

Kim, D.H., Kim, Y.J., Hur, D.S., Jeon, H.S., & Lee, C.H. (2010). Calculating Expected Damage of Breakwater Using Artificial

Neural Network for Wave Height Calculation. Journal of Korean Society of Coastal and Ocean Engineers, 22(2), 126−132.

Kim, H.I., & Kim, B.H. (2020). Analysis of Major Rainfall Factors Affecting Inundation Based on Observed Rainfall and Random Forest. Journal of the Korean Society of Hazard Mitigation, 20(6), 301−310. https://doi.org/10.9798/KOSHAM.2020.20.6.301

Kim, T., Kwon, S., & Kwon, Y. (2021). Prediction of Wave Transmission Characteristics of Low-Crested Structures with Comprehensive Analysis of Machine Learning. Sensors, 21(24), 8192. https://doi.org/10.3390/s21248192

Kim, Y.E., Lee, K.E., & Kim, G.S. (2020). Forecast of Drought Index Using Decision Tree Based Methods. Journal of the Korean Data & Information Science Society, 31(2), 273−288. https://doi.org/10.7465/jkdi.2020.31.2.273

Koc, M.L., Balas, C.E., & Koc, D.I. (2016). Stability Assessment of Rubble-mound Breakwaters Using Genetic Programming. Ocean Engineering, 111, 8−12. https://doi.org/10.1016/j.oceaneng.2015.10.058

Koo, M.H., Park, E.G., Jeong, J., Lee, H.M., Kim, H.G., Kwon, M.J., ... Jo, S.B. (2016). Applications of Gaussian Process Regression to Groundwater Quality Data. Journal of Soil and Groundwater Environment, 21(6), 67−79. https://doi.org/10.7857/JSGE.2016.21.6.067

Kuntoji, G., Manu, R., & Subba, R. (2020). Prediction of Wave Transmission over Submerged Reef of Tandem Greakwater Using PSO-SVM and PSO-ANN Techniques. ISH Journal of Hydraulic Engineering, 26(3), 283−290. https://doi.org/10.1080/09715010.2018.1482796

Lee, G.H., Kim, T.G., & Kim, D.S.(2020). Prediction of Wave Breaking Using Machine Learning Open Source Platform. Journal of Korean Society Coastal and Ocean Engineers, 32(4), 262−272. https://doi.org/10.9765/KSCOE.2020.32.4.262

Lee, J.S., & Suh, K.D. (2020). Development of Stability Formulas for Rock Armor and Tetrapods Using Multigene Genetic Programming. Journal of Waterway, Port, Coastal, and Ocean Engineering, 146(1), 04019027. https://doi.org/10.1061/(ASCE)WW.1943-5460.0000540

Lee, S.B., & Suh, K.D. (2019). Development of Wave Overtopping Formulas for Inclined Seawalls using GMDH Algorithm. KSCE Journal of Civil Engineering, 23, 1899−1910. https://doi.org/10.1007/s12205-019-1298-1

Lee, T.L. (2004). Back-propagation Neural Network for Long-term Tidal Prediction. Ocean Engineering, 31(2), 225−238. https://doi.org/10.1016/S0029-8018(03)00115-X

Li, B., Yin, J., Zhang, A., & Zhang, Z. (2018). A Precise Tidal Level Prediction Method Using Improved Extreme Learning Machine with Sliding Data Window. In 2018 37th Chinese Control Conference (CCC), 1787−1792. http://doi.org/10.23919/ChiCC.2018.8482902

Liu, Y., Esan, O. C., Pan, Z., & An, L. (2021). Machine Learning for Advanced Energy Materials. Energy and AI, 3, 100049. https://doi.org/10.1016/j.egyai.2021.100049

López, I., Aragonés, L., Villacampa, Y., Serra, J.C., (2017). Neural Network for Determining the Characteristic Points of the Bars. Ocean Engineering, 136, 141−151. https://doi.org/10.1016/j.oceaneng.2017.03.033

Mahjoobi, J., Etemad-Shahidi, A., & Kazeminezhad, M.H. (2008). Hindcasting of Wave Parameters Using Different Soft Computing Methods. Applied Ocean Research, 30(1), 28−36. https://doi.org/10.1016/j.apor.2008.03.002

Montaño, J., Coco, G., Antolínez, J.A.A., Beuzen, T., Bryan, K.R., Cagigal, L., ... Vos, K. (2020). Blind Testing of Shoreline Evolution Models. Scientific Report, 10, 2137. https://doi.org/10.1038/s41598-020-59018-y

Na, Y.Y., Park, J.G., & Moon, I.C. (2017). Analysis of Approval Ratings of Presidential Candidates Using Multidimensional Gaussian Process and Time Series Text Data. Proceedings of the Korean Operations Research And Management Society, Yeosu, 1151-1156.

Najafzadeh, M., Barani, G.-A., & Kermani, M.R.H. (2014). Estimation of Pipeline Scour Due to Waves by GMDH. Journal of Pipeline Systems Engineering and Practice, 5(3), 06014002. https://doi.org/10.1061/(ASCE)PS.1949-1204.0000171

Oh, J., & Suh, K.-D. (2018). Real-time Forecasting of Wave Heights Using EOF-wavelet-neural Network Hybrid Model. Ocean Engineering, 150, 48−59. https://doi.org/10.1016/j.oceaneng.2017.12.044

Panizzo, A., & Briganti, R. (2007). Analysis of Wave Transmission Behind Low-crested Breakwaters Using Neural Networks. Coastal Engineering, 54(9), 643−656. https://doi.org/10.1016/j.coastaleng.2007.01.001

Park, J.S., Ahn, K.M., Oh, C.Y., & Chang, Y.S. (2020). Estimation of Significant Wave Heights from X-Band Radar Using Artificial Neural Network. Journal of Korean Society Coastal and Ocean Engineers, 32(6), 561−568. https://doi.org/10.9765/KSCOE.2020.32.6.561

Passarella, M., Goldstein, E.B., De Muro, S., Coco, G. (2018). The Use of Genetic Programming to Develop a Predictor of Swash Excursion on Sandy Beaches. Natural Hazards and Earth System Sciences, 18, 599−611. https://doi.org/10.5194/nhess-18-599-2018

Rigos, A., Tsekouras, G.E., Chatzipavlis, A., & Velegrakis, A.F. (2016). Modeling Beach Rotation Using a Novel Legendre Polynomial Feedforward Neural Network Trained by Nonlinear Constrained Optimization. In L. Iliadis, I. Maglogiannis (Eds.), Artificial Intelligence Applications and Innovations. AIAI 2016. IFIP Advances in Information and Communication Technology, 475, 167−179.

Salehi, H., & Burgueño, R. (2018). Emerging Artificial Intelligence Methods in Structural Engineering. Engineering Structures, 171, 170−189. https://doi.org/10.1016/j.engstruct.2018.05.084

Shahabi, S., Khanjani, M., & Kermani, M.H. (2016). Significant Wave Height Forecasting Using GMDH Model. International Journal

of Computer Applications, 133(6), 13-16.

Shamshirband, S., Mosavi, A., Rabczuk, T., Nabipour, N., & Chau, K.W. (2020). Prediction of Significant Wave Height; Comparison Between Nested Grid Numerical Model, and Machine Learning Models of Artificial Neural Networks, Extreme Learning and Support Vector Machines. Engineering Applications of Computational Fluid Mechanics, 14(1), 805-817. https://doi.org/10.1080/19942060.2020.1773932

Van der Meer, J.W. (1988). Rock Slopes and Gravel Beaches under Wave Attack (Ph.D. thesis). Delft University of Technology, Delft Hydraulics Report, 396.

Van Gent, M.R.A., Van den Boogaard, H.F.P., Pozueta, B., & Medina, J.R. (2007). Neural Network Modelling of Wave Overtopping at Coastal Structures. Coastal Engineering, 54(8), 586-593. https://doi.org/10.1016/j.coastaleng.2006.12.001

Wilson, K.E., Adams, P.N., Hapke, C.J., Lentz, E.E., & Brenner, O. (2015). Application of Bayesian Networks to Hindcast Barrier Island Morphodynamics. Coastal Engineering, 102, 30-43. https://doi.org/10.1016/j.coastaleng.2015.04.006

Yagci, O., Mercan, D.E., Gigizoglu, H.K., & Kabdasli, M.S. (2005). Artificial Intelligence Methods in Breakwater Damage Ratio Estimation. Ocean Engineering, 32(17-18), 2088-2106. https://doi.org/10.1016/j.oceaneng.2005.03.004

Yoon, H.-D., Cox, D.T., & Kim, M. (2013). Prediction of Time-dependent Sediment Suspension in the Surf Zone Using Artificial Neural Network. Coastal Engineering, 71, 78-86. https://doi.org/10.1016/j.coastaleng.2012.08.005

Zanuttigh, B., Formentin, S.M., & Van der Meer, J.W. (2014). Advances in Modelling Wave- structure Interaction Tthrough Artificial Neural Networks. Coastal Engineering Proceedings, 1(34), 693.

Zanuttigh, B., Formentin, S.M., & Van der Meer, J.W. (2016). Prediction of Extreme and Tolerable Wave Overtopping Discharges Thorough an Advanced Neural Network. Ocean Engineering, 127, 7-22.

Author ORCIDs

Author name	ORCID
Kim, Taeyoon	0000-0002-5060-5302
Lee, Woo-Dong	0000-0001-7776-4664

Appendix

Table A1 Detail informations of the reference

Reference	Input variables	Output variables	ML model	Hyper parameter
		Wave prediction		
James et al. (2018)	1) Wave conditon (H_s, T_z and D) 2) Oven-currents (357 values each for u, v) 3) Wind files (12 values each for u and v)	1) Significant wave height (H_s) 2) Wave period (T_z)	ANN SVM	NHN = 20 AF = ReLU
Shmshirband et al. (2020)	1) Surface wind speed	1) Significant wave height (H_s)	ANN SVM ELM	AF = Sigmoid OP = Levenberg-Marquardt
Chen et al. (2021)	1) Significant wave height (H_s) 2) Mean wave direction (D) 3) Wave period (T_z) 4) Peak period (T_p)	1) Significant wave height (H_s) 2) Wave period (T_z) 3) Peak period (T_p)	RF	-
		Tide level		
Lee et al. (2004)	1) 69 tidal constituent ($\cos(w_n t_j)$, $\sin(w_n t_j)$)	1) Tidal levels	ANN	NHN = 7 AF = Sigmoid Momentum factor=0.8
Granta and Nunno (2021)	1) Astronomical tide (AT) 2) Wind speed (WS) 3) Barometric pressure (BP) 4) Observed tide levels (Z-24 ~ Z-1)	1) Tide level	M5P ANN RF	-
		Design variables		
Kim and Park (2005)	1) Permeability of breakwater (P) 2) The number of wave attack (NW) 3) Damage level (S_d) 4) Slope of structure ($\cos\alpha$) 5) Significant wave height (H_s) 6) Wave period (T_s) 7) Dimensionless water depth (h/H_s) 8) Spectral Shape (SS)	1) Stability number (Ns)	ANN	NHN = 12 AF = Non linear

Table A1 Detail informations of the reference (Continuation)

Reference	Input variables	Output variables	ML model	Hyper parameter
Etemad-Shahidi and Bonakda (2009)	1) Permeability of breakwater (P) 2) The number of wave attack (NW) 3) Damage level (S_d) 4) Surf similarity coefficient (ξ_m) 5) Dimensionless water depth (h/H_s)	1) Stability number (Ns)	M5'	-
van Gent et al. (2007)	1) Significant wave height at the structure toe($H_{mo,t}$) 2) Spectral wave period at the structure toe($T_{m-1,\ o}$) 3) Wave obliquity (β) 4) Toe submergence (h_t) 5) Slope of structure ($\cos\alpha$) . . .	1) Wave overtopping discharge (q*)	ANN	NHN = 20 AF = Non linear Bootstrap resampling
Zanuttigh et al. (2016)	14 variables 1) Wave steepness ($H_{m0,t}/L_{m-1,0,t}$) 2) Wave obliquity (β) 3) Shoaling parameter ($h/L_{m-1,0,t}$) 4) Effect of the toe submergence ($ht/H_{m-1,0,t}$) 5) Effect of the toe width ($Bt/L_{m-1,0,t}$) . . .	1) Wave overtopping discharge (q*)	ANN	NHN = 20 AF = Hyperbolic tangent sigmoid OP = Levenberg-Marquardt Bootstrap resampling
Den Bieman et al. (2021)	15 variables 1) Crest Freeboard(R_c) 2) Roughness factor for cot α_u (γ_{fu}) 3) Crest width (G_c) 4) Berm width (B) 5) Slope of structure ($\cos\alpha$) . . .	1) Wave overtopping discharge (q*)	XG boost	Max_depth = 7 Min-child-weight = 5 Learning_rate = 0.05 Subsample = 1 Reg_lamda = 1 Bootstrap resampling
Formentin et al. (2017)	16 variables 1) Wave steepness ($H_{m0,t}/L_{m-1,0,t}$) 2) Wave obliquity(β) 3) Shoaling parameter ($h/L_{m-1,0,t}$) 4) Effect of the toe submergence ($ht/H_{m-1,0,t}$) 5) Effect of the toe width ($Bt/L_{m-1,0,t}$) . . .	1) Wave overtopping discharge (q*) 2) Wave reflection coefficient (Kr) 3) Wave transmission coefficient (Kt)	ANN	NHN = 20 AF = Hyperbolic tangent sigmoid OP = Levenberg-Marquardt Bootstrap resampling
Kuntoji et al. (2018)	15 variables 1) Relative wave steepness (H_i/gT^2) 2) Relative spacing (X/d) 3) Stability number (H_i/Dn_{50}) 4) Relative crest widths (B/d) 5) Relative crest widths (B/L_o) 6) Relative crest heights (h/d) 7) Relative submergence (F/H_i) 8) Relative water depth (d/gT^2)	1) Wave transmission coefficient (Kt)	ANN SVM	1) ANN NHN = 3 2) SVM C = 183.78 ε = 0.0000538 d = 3
Gandomi et al. (2020)	1) Relative chamber width (B/h) 2) Relative rockfill height (d/h) 3) Relative chamber width in terms of wavelength (B/L_p) 4) Wave steepness (H_s/L_p) 5) Wave number multiplied by water depth (kh) 6) Relative wave height in terms of rockfill height (H_s/d) 7) Permeability of the back wall (p)	1) Wave reflection coefficient (Kr) 2) Wave transmission coefficient (Kt)	LR SVM GPR GP ANN	1) GPR Kernel Function = Exp Kernel Scale = 1.664473 Basic Function = Constant SSD = 0.063, 0.105 Sigma = 0.063, 0.105

Table A1 Detail informations of the reference (Continuation)

Reference	Input variables	Output variables	ML model	Hyper parameter
Kim et al. (2021)	1) Relative freeboard (R_c/H_0) 2) Relative crest width (B/H_0) 3) Surf similarity parameter (ξ) 4) Relative crest width (B/L_0) 5) Relative freeboard to water depth ratio (R_c/h) 6) Ratio of the nominal diameter to the crest height (Dn_{50}/h_c) 7) Relative structure height (h_c/h)	1) Wave transmission coefficient (Kt)	GPR ANN GBR RF SVM LR . . .	-
Morphological and morphodynamic				
Hashemi et al. (2010)	1) Min wind speed 2) Wind direction 3) Number of successive wind 4) Number of wind 5) Significant wave height 6) Significant wave period 7) Direction of wave 8) Angle of beach 9) Wind duration	1) Elevation of 10points on each profile 2) Area under each profile curve 3) Length of profile	ANN	NHN = 20 AF = tanh
Rigos et al. (2016)	1) Freeboard (d) 2) inshore slope (ω_1) 3) offshore slope (ω_2) 4) reef width (w) 5) Significant wave height (H_s) 6) Peak wave period (T_p)	1) The distance from the reef top point to the shoreline (y)	ANN	NHN = 4 AF = Legendre polynomial
López et al. (2017)	1) Month of survey profile 2) Steepness corresponding to the maximum wave height 3) H_{max} direction 4) Days elapsed from H_{max} to the survey profile 5) H_m 6) d_{50} 7) Difference in beach width between profiles	1) Distance from shoreline to the start of the bar (X_s) 2) Depth of the starting point of the bar (Y_s) 3) Distance from shoreline to the crest (X_c) 4) Depth of the crest (Y_c) 5) Distance from shoreline to the final of the bar (X_f) 6) Final point depth (Y_f)	ANN	NHN = 12 AF = sigmoid

NHN = Number of Hidden Neuron, AF = Activation Function, OP = Optimizer