



Data-driven approaches in predicting scour depths at a vertical seawall on a permeable shingle foreshore

M. Salauddin¹ · D. Shaffrey¹ · M. A. Habib¹

Received: 29 November 2022 / Revised: 27 April 2023 / Accepted: 27 April 2023 / Published online: 5 May 2023
© The Author(s) 2023

Abstract

Despite recent research efforts in advancing machine learning (ML) tools to predict nearshore characteristics at sea defences, less attention has been paid to ML algorithms in predicting scouring characteristics at vertical seawalls. In this study, four ML approaches were investigated, including gradient boosting decision trees (GBDT), random forest (RF), support vector regression (SVR), and ridge regression (RR). These approaches were utilised to predict scour depths at the toe of an impermeable vertical seawall in front of a permeable shingle slope. The developed ML algorithms were trained and tested (70% for training and 30% for testing) using the scouring datasets collected from laboratory tests performed on seawalls in a 2D wave flume at the University of Warwick. A novel hyperparameter tuning analysis was performed for each ML model to tailor the underlying dataset features while mitigating associated data overfitting risks. Additionally, the model training process demonstrated permutation feature importance analysis to reduce overfitting and data redundancy. The model predictions were compared with the observed values using the coefficient of determination (R^2) score, root mean square error (RMSE), and Pearson correlation R -value. Consequently, the RF and GBDT methods accurately predicted scour depths at the toe of vertical seawalls with shingle foreshores. This study produced data, information, and a model that could directly or indirectly benefit coastal managers, engineers, and local policymakers. These benefits included forecasting scour depths and assessing the impact on the structural integrity of the sea defences in response to the threat imposed by extreme events, which are essential for the sustainable management of coastal protections and properties behind such structures in coastal areas.

Keywords Climate resilience · Gradient boosting decision trees · Random forest · Scouring · Vertical seawalls

Introduction

Toe scouring is described when sediment materials are removed from underneath the toe of the sea defence structures (scour holes). Typically, this phenomenon occurs due to the erosive wave action around the seabed. The formation of scour depths at the toe of a structure leads to sea defences (vertical seawalls and breakwaters) failures, which are established to protect the properties and personnel behind these infrastructures. Global sea-level rise concerns associated with climate change are also increasing, with extreme climatic events on various sea defences expected to be more frequent. This impact is a significant threat to the integrity

and functional efficiency of these structures (Dong et al. 2020a, b; O'Sullivan et al. 2020; Salauddin et al. 2021a, b). As such faced with climate change and rising waters, we must adapt our coasts to protect hinterland properties, infrastructures, and populations. Thus, reliable prediction of scour depths at sea defences is crucial to minimise the risks to human assets owing to climate change while ensuring the effective long-term management of coastal protections as well as nearby coastal areas.

Several empirical prediction tools (Fowler 1992; Sutherland et al. 2003, 2006; Wallis et al. 2009; Salauddin & Pearson 2019a, b) were discovered in estimating the toe scour depths at vertical breakwaters based on small-scale laboratory measurements in a controlled environment. Nevertheless, scouring is produced from the complex interaction between a structure and sediment. The accuracy of these empirical methods in predicting scour depths is often limited due to the low representation accuracy of wave structure interactions within the experimental settings (Pourzangbar

✉ M. Salauddin
md.salauddin@ucd.ie

¹ School of Civil Engineering, UCD Earth Institute and UCD Dooge Centre for Water Resources Research, University College Dublin, Dublin 4, Ireland

et al. 2017a). Each experiment only investigates the influence of one or two critical parameters on scouring patterns. Additionally, physical model experiments are expensive and time-consuming, particularly in new laboratory model constructions with complex geometrical configurations.

Several machine learning (ML) algorithms have been extensively used in recent years to predict critical coastal processes at sea defences widely. These MLs include overtopping (Habib et al. 2022a, b; den Bieman et al. 2021a,b), wave runup (Abolfathi et al. 2016), and scouring (Pourzangbar et al. 2017a,b). Although extensive research has been performed on employing data-driven techniques in predicting coastal processes, only several studies on scouring predictions at seawalls are observed. For example, Pourzangbar et al. successfully employed two data-driven approaches, such as genetic programming (GP) and artificial neural networks (ANNs), to predict maximum scour depth at breakwaters on sandy foreshores subjected to non-breaking waves (Pourzangbar et al. 2017a). The recent developments in deriving artificial intelligence-based prediction tools for scouring predictions have also observed applications of other ML approaches, such as support vector regression (SVR) and model tree algorithm (M5'). In 2017, Pourzangbar et al. demonstrated an SVR and M5' ML model in predicting maximum scour depth at a breakwater on a sandy bed using the dataset available in the literature (Pourzangbar et al. 2017b). Moreover, the study examined the efficiency of such data-driven approaches in forecasting scour depths compared to those reported by existing empirical prediction tools, such as Pourzangbar et al.'s work (Pourzangbar et al. 2017a,b). When these studies were analysed, the findings suggested that ML algorithms could model and predict complex wave-structure interactions, such as toe scouring at coastal flood defences.

To date, a research aspect mainly focusing on the developed scouring patterns for beaches with fine sediments, such as sandy beaches, is reported by multiple coastal protection studies on sandy beds (Xie 1981; Fowler 1992; Sumer & Fredsøe, 2000; Sutherland et al. 2003; 2008; Müller et al. 2008; Gislason et al. 2009; Wallis et al. 2009). Comparatively, these studies are more focused than higher coarse grain sediment, such as shingle beaches described by several coastal protection studies on shingle beds (Powell 1990; Powell & Lowe 1994; Salauddin and Pearson 2019a). To the authors' knowledge, no advanced ML tool development studies to predict scour depths on sea defences with shingle or gravel beaches were investigated. Although data-driven ML approaches have gained popularity owing to their ease of use with robust prediction performance, in-depth knowledge of the accuracy and efficiency of such techniques is still required to estimate scour depths at seawalls.

In this study, the performance of data-driven techniques (ML algorithms), such as gradient boosting decision trees

(GBDT), random forests (RF), and SVR, was evaluated to predict scour depths at vertical seawalls with shingle foreshores. The study was investigated based on the training and testing of the recently collected experimental datasets. In addition to these three ML methods, a linear regression model, such as ridge regression (RR), was successfully applied as a control measure to demonstrate the effectiveness of the three non-linear ML models.

Materials and Methods

Scouring Dataset

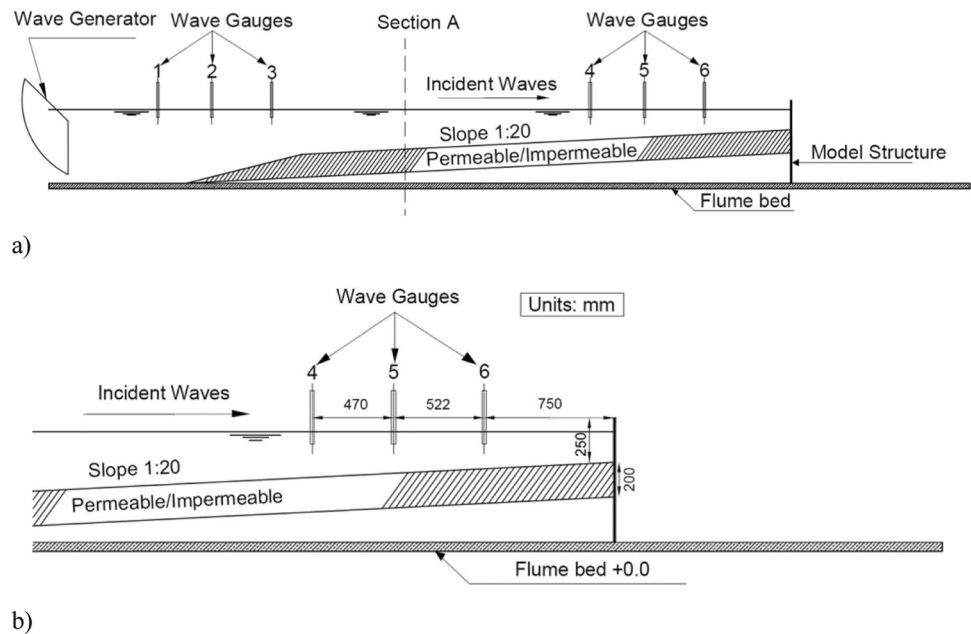
The physical model measurements reported by Salauddin and Pearson were considered in this study, which was conducted in a 2D wave flume (22 m long, 0.6 m wide, and 1 m deep) with a geometric scale of 1(V):50(H) (Salauddin and Pearson, 2019b) in the University of Warwick. The wave channel was equipped with a piston-type wave paddle capable of generating both regular and irregular sea states. Furthermore, the small-scale laboratory experiments were performed on a plain vertical seawall in front of a smooth permeable 1 in 20 foreshore slope (Salauddin & Pearson 2018, 2019b), covering a wide range of incident wave conditions for impulsive and non-impulsive waves. The incident waves were subsequently generated using a JONSWAP wave spectrum with a peak-enhancement factor of 3.3, representing the young sea states. Six different toe water depths were tested, with the relative crest freeboards (R_c/H_{m0}) varied from 0.5 to 5.0. Approximately 120 experiments were performed on plain vertical seawalls to predict the scouring characteristics of impulsive and non-impulsive wave attacks.

The wave measurement techniques developed by Salauddin and Pearson (2019a, b) were employed in other wave flume investigations on seawalls (Dong et al., 2018, 2021; Salauddin et al. 2020). Hence, the scour depths in front of the structure and several locations along the wave flume were recorded after a wave attack. The maximum scour depth for a given storm was derived from the measured scour depths along the entire foreshore length. Based on the measurements, the maximum scour depth was achieved in front of the seawall for all test cases. Additionally, the reader was referred to Salauddin et al.'s work for information on the laboratory test setup and measurements (Salauddin and Pearson 2018, 2019a, b). Figure 1 illustrates the experimental setup of the study proposed by Salauddin and Pearson (2019a).

Modelling Approaches

Within the summary statistics of a dataset, ML is a branch of computer science that identify patterns or relationships while generating a predictive model (Kuhn &

Fig. 1 Laboratory setup for scouring experiments at an impermeable seawall. (a) A layout of the test setup, and (b) the position of wave gauges. Adopted from Salauddin & Pearson 2019a



Johnson 2013). Three main ML approaches are identified: supervised, unsupervised, and reinforcement learning. In supervised learning, the labelled input and output data are functionally related to predicting outcomes. In contrast, unsupervised learning models discover the inherent structure of unlabelled data. Meanwhile, supervised learning algorithms are of classification and regression types (Kuhn & Johnson 2013). The classification and regression algorithms are used to predict a label (binary or multi-class) and produce a predictive model for continuous quantities, respectively. This work utilised the k-fold cross-validation and hyperparameter tuning approaches to optimise an ML-based model.

The difficulty in choosing the most effective ML method to build a predictive model was emphasised in a study by Pourzangbar et al. (Pourzangbar et al. 2017a). Depending on the specific dataset, there were inherent advantages and disadvantages of using one model over the other. When building any predictive model using ML methods, several different ML methods were suggested to be investigated as there was often no apparent reason one model would perform better. A recent systematic literature review by Habib et al. reported that the most common ML approaches for predicting key coastal processes, including overtopping, scouring, and wave runup, were M5', RF, GBDT, SVR, and ANNs (Habib et al. 2022a). For example, Habib et al. revealed that decision tree (DT) algorithms (M5' and GBDT) and ANNs could perform regression to produce predictive models for overtopping and associated parameters (Habib et al. 2022a). On the

contrary, GBDT, RF, and SVR were adopted as their algorithms in this study, which examined their performances in predicting scouring depth at vertical seawalls.

GBDT

In this study, the regression-based GBDT algorithm was applied. The primary function of a DT is to train on a given set of vectors or parameters. These data are then repeatedly split until they have achieved the same labels. The split is produced based on certain thresholds, which the DT learns after training on the training data. Following the training, the algorithm predicts the label of a test dataset. Thus, DTs are utilised for classification (predicting discrete labels) and regression (predicting continuous labels). The split or decision rules for regression tasks are based on the mean square error between the actual and predicted labels.

The DT application is backed by the simplicity and ability to perform training on datasets with missing algorithms values (Pedregosa et al. 2011). The difference between the actual and predicted values is expressed as a loss function. Recent advancements in the application of ML models depicted that a more robust DT approach, such as GBDT, is adopted to reduce unnecessary overfitting on the training data. A GBDT model effectively reduces the value of the loss function to the minimum through subsequent iterations of training and testing. Hence, this reduction improves the accuracy of the predictions while keeping overfitting in check (Sutton 2005). The GBDT method is utilised as follows:

1. Define configurational inputs (differentiable loss function) and the investigated dataset in the form of $\{(x_i, y_i)\}_{i=1}^n$, where x_i are the predictor variables and y_i is the outcome variable for n samples or data rows with $i \in n$. The function $F(x_i)$ represents the estimation of the outcome variable.
2. Determine an appropriate initial prediction for the output variable that minimises the loss function of $(\Psi(y_i, F(x_i)))$. This prediction of the outcome variable is denoted as γ as follows:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n \Psi(y_i, \gamma) \tag{1}$$

3. For $k = 1$ to K , where K is the number of iterations, do:
 - i. Compute the pseudo-residual for each sample ($r_{i,k}$) by evaluating the gradient at the previous prediction of the outcome variable ($F_{k-1}(x_i)$):

$$r_{i,k} = - \left[\frac{\partial L(\Psi(y_i, F(x_i)))}{\partial F(x_i)} \right]_{F(x_i)=F_{k-1}(x_i)} \tag{2}$$

- ii. Fit a DT to the computed pseudo-residuals and determine terminal regions ($R_{j,k}$), which predicts a separate constant value where j is the number of leaves in each terminal region.
 - iii. Produce a new prediction for each sample using the DT fitted in the previous step and the previous prediction. This prediction is achieved by finding the value γ that minimises the loss function:

$$\gamma_{i,k} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{j,k}} \Psi(y_i, F_{k-1}(x_i) + \gamma) \tag{3}$$

- iv. Update the predicted values of each sample by adding the newly predicted model to the previously predicted model. Furthermore, sum the output values ($\gamma_{j,k}$) for all the terminal regions ($R_{j,k}$) with a weighting that is given by the learning rate v :

$$F_k(x) = F_{k-1}(x) + v\gamma_{j,k}I_j(x \in R_{j,k}) \tag{4}$$

where I_j is a $j \times j$ identity matrix.
end for:

- v. Output and summarise final predictions for $F_K(x)$.

In Step 2 of Algorithm 1, the squared error of residuals multiplied by half is a pragmatic choice of loss function as it simplifies the loss function minimisation. The derivative of the loss function becomes the residual of the specific

sample under investigation, thus leading to the average observational value as the initial prediction.

SVR

Support vector machines (SVMs) are one of the popular statistical methods typically used for classification. Therefore, SVMs define an appropriate threshold within a given dataset to segregate the data. The data points on this threshold are classified as one group, while the other is classed as another. Subsequently, the support vector classifiers account for outliers in the dataset by tolerating misclassification and for a certain degree of overlapping classifications. This method is extended for regression by modifying the algorithm to take continuous quantities. Thus, this algorithm is known as SVR.

The main hyperparameters for SVR refer to the type of kernel used with cost and gamma parameters. The kernel type specifies the kernel function of the SVR algorithm to perform regression tasks. Moreover, the default kernel type in the Scikit-learn library is the radial basis function. This kernel type is very effective and a reasonable default kernel type in many common scenarios (Kuhn & Johnson 2013; Buitinck et al. 2013). The cost parameter, or c parameter, controls the complexity of the model by penalising misclassified data points. Hence, excessively high c parameter values lead to overfitting the model, which heavily penalises misclassification in the training set. Resultantly, the model cannot perform as well on unseen data. Alternatively, extremely low c parameter values produce model underfitting as more misclassification is tolerated. Furthermore, the ability of the model to learn the training data structure is compromised. The gamma parameter controls the boundary around data points represented in higher dimensions. They are grouped if a data point is within the boundary of another data point. Consequently, lower or higher gamma lead values lead to specific or generalised boundaries, respectively.

RF

Bootstrapping is another resampling technique that reduces how much a model overfits the training dataset (Bishop 2006). A bootstrap dataset is generated by taking samples from the dataset with replacement. This dataset generation indicates that the same samples are selected more than once from the original dataset. The bootstrap dataset is the same size as the original dataset, as the specific samples will likely appear multiple times in the dataset. In contrast, other samples do not appear. The samples from the original dataset that do not appear in the bootstrap dataset are also termed out-of-bag samples. Therefore, bootstrapping is typically performed over many iterations, with each iteration generating a new bootstrap dataset and producing a model from

this bootstrap dataset (Kuhn & Johnson 2013). The results from these iterations are collectively assessed in a process known as bagging. Although a common method for building these models is to utilise DTs, potential bias is introduced in the DTs as all trees are only partially independent of each other. This dependency between DTs is named tree correlation, which improves the bagging performance with this dependency reduction (Kuhn & Johnson 2013). For example, randomness is introduced into the models when DTs are produced.

An RF model commonly uses an ensemble of DTs. In each DT, predictors are selected randomly after each DT split, which lessens tree correlation. Finally, the model generates a prediction, averaged across the models, to provide the overall prediction. The number of predictors selected at each DT split is the main tuning parameter for RF models. The RF is more computationally efficient than the bagging approach as fewer predictors are evaluated at the DT split (Kuhn & Johnson 2013). The general algorithm for an RF is demonstrated as follows:

1. Build m DT models.
2. For $i = 1$ to m , do:
3. Generate a bootstrap sample dataset from the original dataset.
4. Train a DT model on this generated sample dataset.
5. For $i = 1$ to m , do:
 - i. Randomly select k original predictors, where k is less than the number of original predictors.
 - ii. Select the best-performing predictor and partition data based on this predictor.

end for:
 - iii. End DT and keep all non-critical and redundant instances.

end for:
 - iv. Return.

RR

A penalty function in RR penalises large weight values and prevents overfitting. This method adds a penalty function to the sum of squared errors (SSE) of the predicted parameters portrayed in Eq. 5 (Kuhn & Johnson 2013) as follows:

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^P \beta_j^2 \quad (5)$$

where y_i is the model outcome, \hat{y}_i is the model prediction, λ is the penalty coefficient, β is a vector that contains the parameter estimates for each predictor for n model predictions, and P is the parameter estimations. The parameter

estimates with large SSEs decrease with the penalty λ . The regularisation is a way of controlling these estimates, in which the penalty function is added to the SSE of parameter estimates. Despite this complexity, the RR approach remains a fundamentally linear model. The final predictive model may not have the capacity to provide a sophisticated description of the structure in the data compared to other methods, such as GBDT or SVR.

k-fold Cross-validation

The resampling method reduces the likelihood of a predictive model being overtrained on the training set by model retraining for several iterations using a different training and testing set for each iteration. Thus, the results from each iteration are summarised at the end of this process. The k -fold cross-validation is a common resampling technique that eliminates bias in predictive models. In k -fold cross-validation, the data is randomly partitioned into k sets of roughly equal size. A model is fit using all samples, excluding the first subset (first fold). Subsequently, the fitted model is validated against the held-out sample to estimate the performance of the model. The first subset is returned to the training set. Then, the procedure is repeated with the second subset held out. This method is repeated k (user-defined parameter) times for each subset.

The k -resampled performance estimates are summarised along with the mean and standard error. Furthermore, the relationship between the tuning parameters and model utility is understood (Kuhn & Johnson 2013). Meanwhile, cross-validation specifically reduces the risk of overfitting and tends to reduce the variance of the model. Thus, the essential features influencing the prediction task can be deduced using a feature importance analysis.

Hyperparameter Tuning

When adequate model tuning for the learner is ensured, the hyperparameter tuning approach is employed to reduce the underfitting likelihood of a model occurring. The process involves picking optimal hyperparameters for the predictive model, in which the learner with sufficient complexity captures the training set structure. Simultaneously, this process is not overly influenced by data noise. Two typical hyperparameter tuning methods are utilised based on Scikit-learn functions: GridSearchCV() and RandomizedSearchCV(). The GridSearchCV() function produces hyperparameter tuning for an ML model by exhaustively combining each parameter listed in the hyperparameter space and fitting a model for each combination. Consequently, GridSearchCV() is often computationally expensive. Furthermore, introducing parameters or broadening existing parameter ranges in the hyperparameter space increases the computational time.

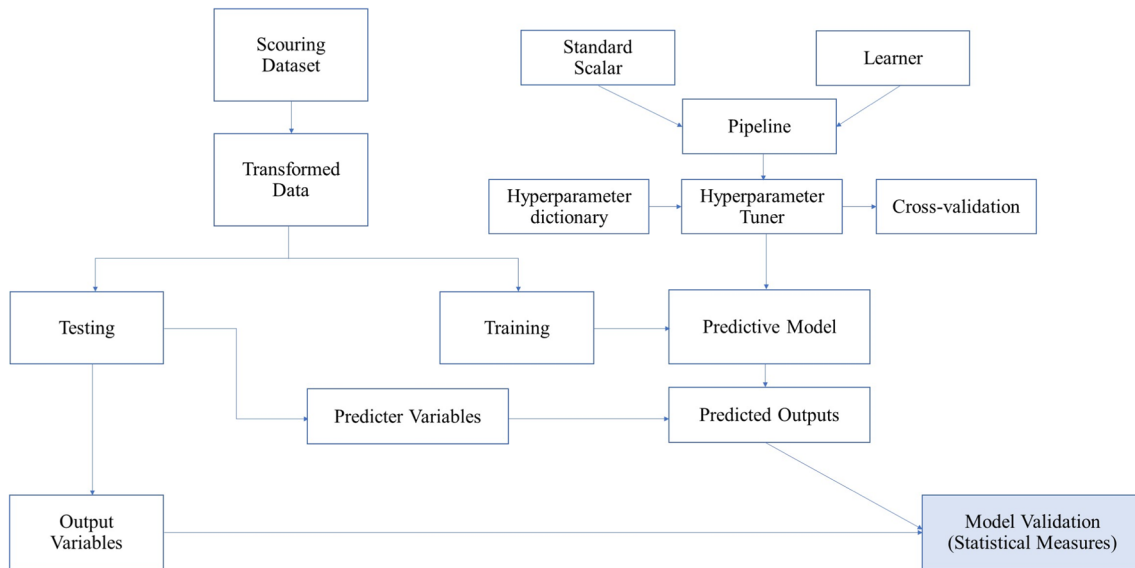


Fig. 2 Schematic flowchart of the methodology utilised in this study

Alternatively, the `RandomizedSearchCV()` function is not computationally expensive as it does not fit a model to every parameter combination in the hyperparameter space. Instead, the `RandomizedSearchCV()` selects a smaller subsample and fits models using the hyperparameter space combinations.

Feature Importance

In predicting the behaviour of the response variable, the feature importance is a method that determines each predictor variable or the importance of the feature. Hence, changes in essential features produce significant changes in the response variable than less critical features. Additionally, the method eliminates unnecessary features from the overall model, which aids in simplifying model interpretation. Permutation feature importance (PFI) is an example of a common type of feature importance. The PFI performs well for models that do not support feature importance by calculating relative importance scores independent of the model used. Therefore, the PFI mechanism is described as follows:

1. Focusing on one feature at a time, the variables are shuffled with a prediction established from this shuffled feature.
2. The variability between the prediction and the actual output is assessed.
3. Prior to shuffling, the feature is returned to its original state.
4. Steps 1 to 3 are repeated for each feature in the dataset.

5. Important features are determined by comparing the individual score of a feature with a mean importance score.

Model Performance Evaluation

The performance of a predictive model assessed with various statistical metrics, such as coefficient of determination (R^2) score, root mean square error (RMSE) and Pearson R -value, were calculated. The R^2 and Pearson R -values indicate to what extent the algorithm has fitted the training data while comparing the RMSE value indicates if there is any outlier in the predicted results. Hence, this study briefly discusses these methods to provide additional background on the exact measurements.

RMSE

The RMSE is a function of residuals for a model, which is the difference between the observed and predicted values (see Eq. 6). Moreover, the RMSE function returns a single measurement of the collective deviation of predicted values from the actual observed values. A lower RMSE represents a better agreement between the predicted and actual values. Therefore, RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad (6)$$

Table 1 Summary of the best hyperparameter combinations for the GBDT approach

Hyperparameters	GridSearchCV()	RandomizedSearchCV()
Max depth of DTs	3, 6, 10, 12	All integers between 1 and 100
Learning rate	0.01, 0.05, 0.1	Numbers between 0 and 10 at 0.05 increments
Number of estimators	50, 100, 500, 1000	All integers between 1 and 2000
Column subsample ratio by tree	0.1, 0.3, 0.7	Numbers between 0 and 1 at 0.05 increments

where x_i and \hat{x}_i are the i^{th} members for the observed and predicted extent of scouring, respectively, of a sample size N .

R^2

The R^2 score measures the correlation between two variables, which evaluates the proportion of data information as explained by the model as follows:

$$R^2 = 1 - \frac{\text{Sum of Squares of Residuals}}{\text{Total Sum of Squares}} \tag{7}$$

Based on Eq. 7, a higher R^2 score indicates a better data explanation by the model ($R^2 = 1$ indicates that all information is attributable to the model). Although the R^2 score is a good metric for the correlation between predicted and actual outcomes, the accuracy of predicted to actual outcomes could not be inferred.

Pearson R -value

The Pearson correlation coefficient or the Pearson R -value is a standard method to measure the linear correlation of predictions. Thus, the method measures the linear correlation between predicted and observed variables as expressed by Eq. 8:

$$\text{Pearson } R = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum(x_i - \bar{x})^2)(\sum(y_i - \bar{y})^2)}} \tag{8}$$

where x_i is the predicted outcome, \bar{x} is the average of predicted outcomes, y_i is the actual outcome, and \bar{y} is the average of actual outcomes.

Table 2 Summary of the hyperparameter combinations for the SVR approach

Hyperparameters	GridSearchCV()	RandomizedSearchCV()
C parameter	0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 100	0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 100
Learning rate	0.0001, 0.001, 0.01, 0.1, 0.2, 0.3	0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

Residual Plots

Although the measured statistical error for predicted models provides valuable insights into the underlying structure and relationships present in a dataset, the noise present in the dataset remains challenging to distinguish. Detailed data visualisation, such as residual plots (difference between the predicted and observed values) of predictive models and statistical analysis (Anscombe 1973), should be performed in the thorough investigation of a given dataset. In a predicted model that perfectly simulates the real-world behaviour in each dataset, the residuals have no relationship with either the predictor or the predicted outcome variable. In contrast, if the theoretical description is imperfect, a relationship between the residuals and the predictor variables or the predicted outcome variable is formed.

The strength of this relationship evaluates the performance of the predictive model. Stronger and weaker relationships indicate poorer and better-performing models, respectively. Therefore, the strength of this relationship is illustrated by plotting the residual values against the values for the predicted outcome variable (Anscombe 1973). In this study, the residual values for each predictive model were determined and subsequently plotted against predicted scour depths to evaluate the correlation between residuals and the predictor variables or the predicted outcome variable.

Methodology

Figure 2 depicts the overall methodology flowchart used to build ML models for predicting scour depths. Initially, the raw dataset (Salauddin & Pearson 2019b) was transformed by effectively preparing the input parameters for manipulation and splitting. Hence, this process involved cleaning the dataset by removing blank spaces, null values, and irrelevant fields (directory names). Subsequently,

Table 3 Summary of the best hyperparameter combinations for the RF approach

Hyperparameters	GridSearchCV()	RandomizedSearchCV()
Bootstrap	True	True, False
Maximum Tree depth	5, 10, None	All integers between 1 and 20
Maximum number of features	'auto', 'log2'	'auto', 'log2'
Number of estimators	5,6,7,8,9,10,11,12,13,15	All integers between 1 and 20

the dataset was split into training (70%) and testing (30%) sets from the original data. The training and testing data were further separated into predictor variables or features and the response or output variables. This study utilised the training and test sets for pattern recognition and validation, respectively.

A pipeline was constructed to combine scaling and learner information into a new estimator using a scaling function and a specific learner for the predictor model. A tenfold cross-validation was performed on the training dataset, which split the training data into ten subsets. Each subset validated the pattern recognition before applying it to the test set. The cross-validation step was then repeated thrice. Meanwhile, hyperparameter tuning and cross-validation were adopted to reduce the overfitting of the training data. The hyperparameter tuning essentially curtailed a given dataset for a particular ML algorithm. At the same time, the predictive model employed the features from the testing dataset to predict output variables for each dataset sample. Finally, the predicted output variables were compared with the actual output variables of the testing set.

Results and Discussions

Table 1 tabulates three cases constructed and tested within this study to investigate the optimum configuration for the tested models. Based on Sect. "Methodology", Fig. 2, the overall process of developing predictive models was identical for all test configurations. Thus, the main difference between Test Case 1 and Test Case 2 was the cross-validation repetitions for the latter case. When comparing Test Case 2 and Test Case 3, instead of using GridSearchCV(), Test Case 3 demonstrated the same configurational setup as Test Case 2. In Test Case 3, the hyperparameter tuner RandomizedSearchCV() detected optimal parameters for the developed model. A comprehensive description of each Test Case is outlined in Table 1. Unless otherwise specified, the study results only represented the outcomes corresponding to Test Case 3 for all models. Resultantly, Test Case 3 provided the most satisfactory predicted values compared to other cases.

Hyperparameter Combinations

The hyperparameters for GBDT included factors such as the number of trees in the ensemble system, tree depth, learning rate, number of subsamples used or rows, and number of features. Table 2 outlines the combination of optimum hyperparameters for the constructed GBDT model based on the GridSearchCV() and RandomizedSearchCV() functions.

In the SVR approach, C and γ were the two main hyperparameters controlling the penalisation of misclassification and the coefficient of the kernel function, respectively. An appropriate C value allowed the structure of the dataset to be adequately captured by the model while the γ parameter controlled the level of complexity of the learner. With high or low γ values, the effect of the kernel function was exaggerated or diminished, respectively. Meanwhile, high C and γ values could lead to overfitting, while low C and γ values lead to underfitting on the training set. Additionally, ϵ was the parameter that controls how aggressively the model learned from the training data. Relatively higher values corresponded to a more aggressive learner, whereas lower values corresponded to a less aggressive learner. In this study, Table 3 outlines the optimum hyperparameter spaces constructed with the GridSearchCV() and RandomizedSearchCV() functions of the SVR approach.

The essential hyperparameters for the RF ML model were the number of estimators, number of trees, maximum tree depth, and number of features after each DT split (see Table 4). Similarly, the hyperparameter spaces were

Table 4 Summary of various tested configuration descriptions of the predictive models

Test Case No	Configuration
Test Case 1	- Training-test data split - Data-normalisation - Cross-validation: 10 folds, no repetition - Hyperparameter training: GridSearchCV()
Test Case 2	- Training-test data split - Data-normalisation - Cross-validation: 10 folds, 03 repetitions - Hyperparameter training: GridSearchCV()
Test Case 3	- Training-test data split - Data-normalisation - Cross-validation: 10 folds, 03 repetitions - Hyperparameter training: RandomizedSearchCV()

constructed using the GridSearchCV() and RandomizedSearchCV() functions. The only hyperparameter term for RR to be cognizant of was the α parameter, which controlled the regularisation strength of the model and produced strictly positive values. Hence, the values of the α parameter were integers from 0 to 10 with a 0.1 increment.

Results of Prediction with ML models

The comparison of measured or actual scour depths with predicted values of the tested algorithms is presented in Fig. 3. A linear regression line was represented from the blue line for the plotted data, with the blue band depicting the 95% confidence interval (CI) lines. The data points corresponded to the ML models (GBDT, SVR, and RF) as most of the predicted values cluster were within or close to the confidence interval. This observation indicated that the predicted values correlated efficiently with the measured values. In addition, the prediction from the linear regression model (RR, see Fig. 3d) was more scattered from the regression line and CI than ML models. Therefore, the RR

model was suggested to be the least accurate model for predicting scour depths within this study.

The residual plots for the predictive models describing the relationship between residual values and predicted scour depths are plotted in Fig. 4. The data points for each developed model were randomly distributed, thus indicating a strong correlation between the residuals and predictions for the tested dataset. These results implied that the predictive models performed well, producing higher confidence for these predictions.

Feature Importance

Figure 5 illustrates the heatmap of the correlation among different tested features adopting the approach of Shetye's work (Shetye 2019). From the heatmap, the model features did not reveal a strong linear relationship among them. Inevitably, this result suggested that a linear regression model was inappropriate for predicting features from the used dataset. Furthermore, the result represented that none of the predictor variables depicted a linear relationship with the outcome variable (scour depth).

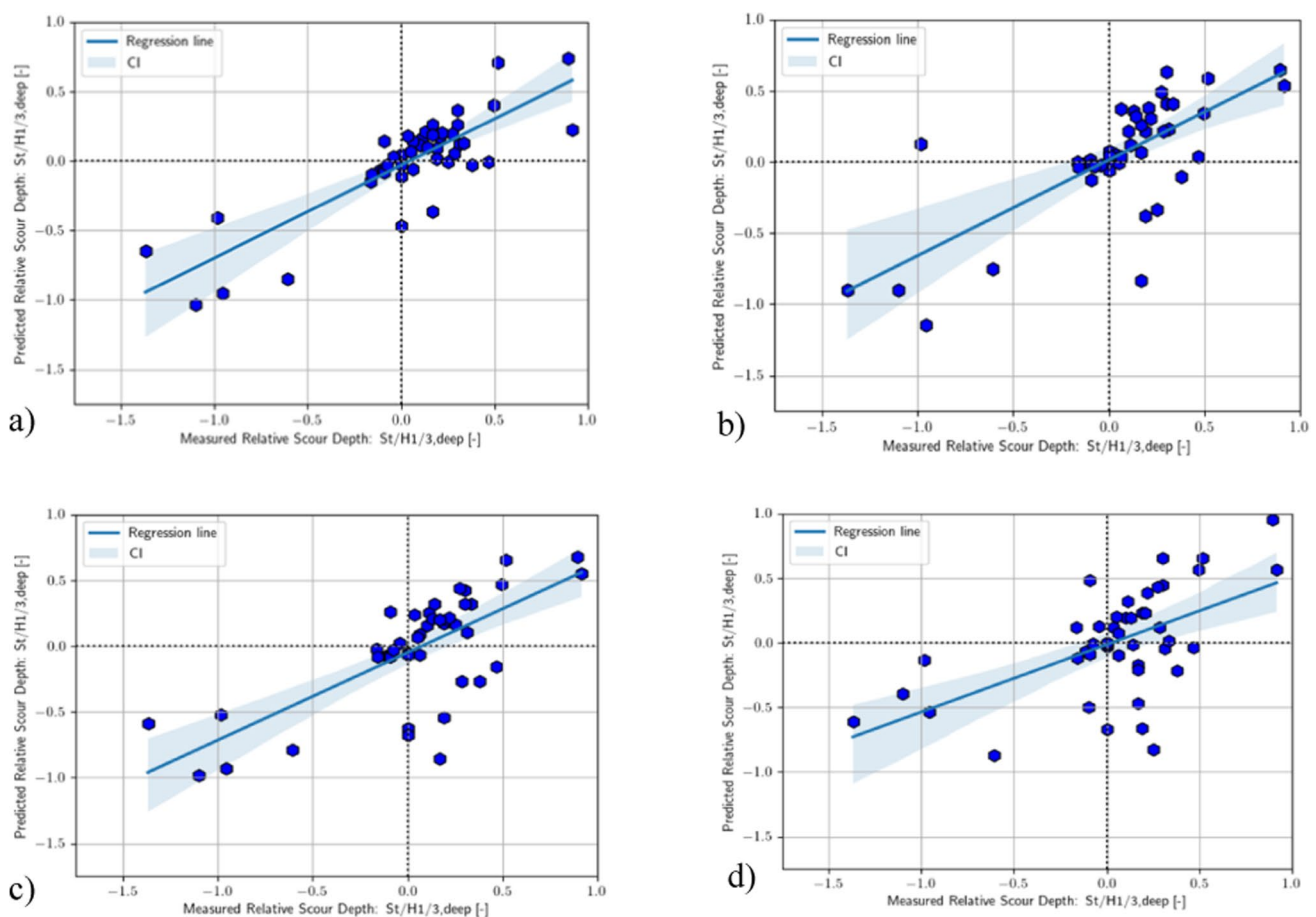
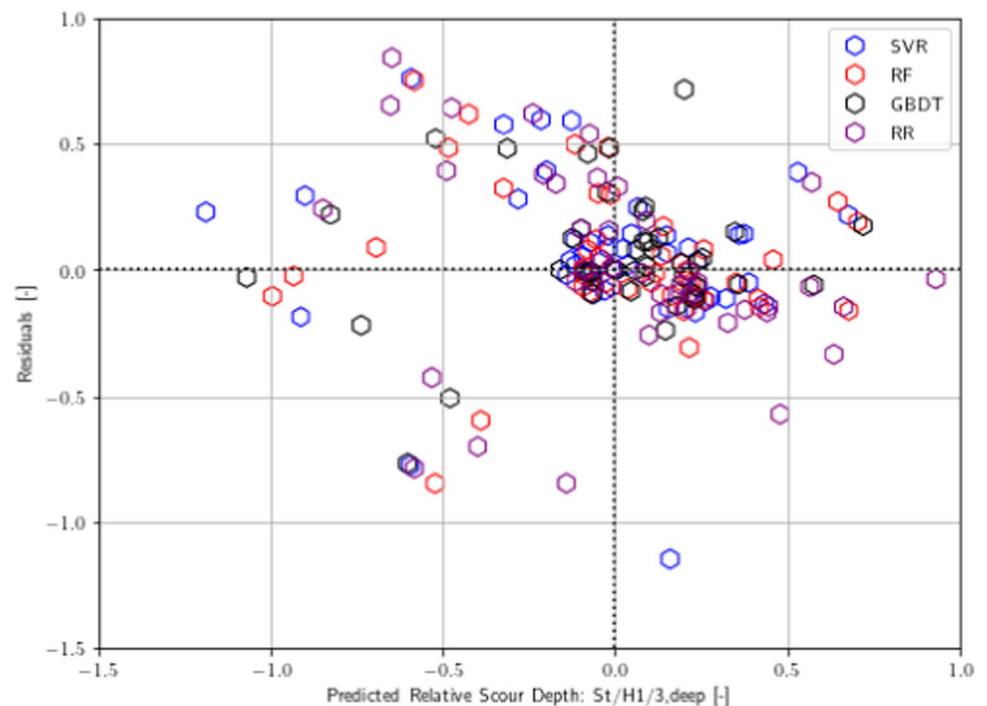


Fig. 3 Predicted versus actual values of scour depth from the four algorithms: (a) GBDT, (b) SVR, (c) RF, and (d) RR

Fig. 4 Residuals from the prediction of the tested algorithms



Although the heatmap served as a holistic metric to evaluate individual model performance, it does not provide insight into the most impactful features. The permutation feature importance analysis was applied to identify the essential features of each model in predicting the extent of scouring. Since they were clear indications that only GBDT and RF models performed well on the vertical seawall, the feature importance results for these models were only discussed. Figure 6 reports the feature importance results for the GBDT and RF models. When comparing the resulting feature importance of the two ML methods, there were several differences between the GBDT and RF models. Surprisingly, the Iribarren number was ranked lesser than expected in both cases. Overall, the results revealed that the relative toe depth, wave height, Iribarren number, and wave impulsiveness parameter were more critical features in predicting the relative scour depths at vertical seawalls. Hence, these findings agreed with the observations of Powell and Salauddin's works for scouring at vertical seawalls with permeable shingle slopes (Powell 1990; Salauddin & Pearson 2019a).

Comparison of GBDT, RF, SVR, and RR Models

The predicted values of ML models were compared with the observed or actual values by determining the statistical error metrics. Table 5 demonstrates that the overall RF and GBDT models effectively forecast the extent of scouring compared to the performance of SVR and RR models. Interestingly, RF also performed slightly better compared to the GBDT approach. The RF model achieved an R^2 value of 0.729 and an

RMSE value of 0.230 for the test set, while the GBDT model achieved an R^2 value of 0.719 and an RMSE value of 0.235 for the test set. A small RMSE value for the RF and GBDT approaches indicated that the collective residuals between the scouring values were relatively small. The impact of minimizing RMSE in ML evaluation is that it encourages the model to make predictions that are as close as possible to the true values. The ML models were designed to perform in a way that minimizes the RMSE, which effectively reduces the difference between the observed and predicted overtopping values (the desired outcome from the ML models). Thus, this indication demonstrated a good agreement between the predicted and observed values. Moreover, the high R^2 and Pearson R -values suggested a strong positive relationship between the predicted and observed extent of scouring.

Another two tested models (SVR and RR) reported a considerable variation in predicted scour depths compared to actual or measured values. This variation was evident from the statistical error metrics corresponding to these models. The comparisons of the SVR and RR model results for the training and test sets with RF and GBDT values described that neither the SVR nor the RR model performed well in scouring predictions. Additionally, the RR model yielded the poorest prediction results among all the models. The prediction indicated that the scouring depth features at the vertical seawall produced a non-linear relationship. At the same time, RR is a fundamentally linear regression algorithm inappropriate for this dataset. Therefore, RF and GBDT were successfully employed to predict scour depths at vertical seawalls with shingle foreshores.

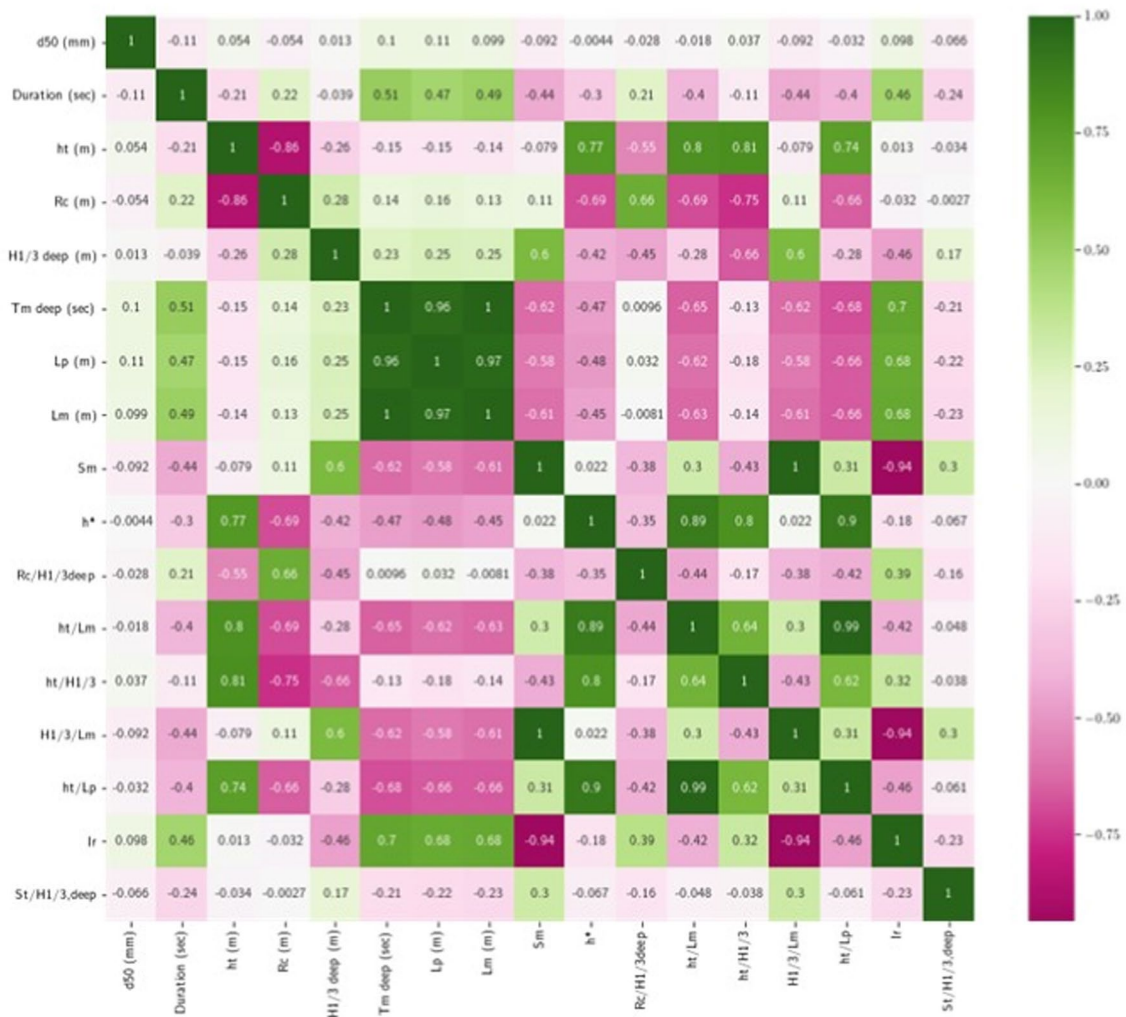


Fig. 5 Heatmap indicating the correlation of the predictor variables and outcome variable for the dataset. The correlation values are demonstrated in the right-most column and bottom row

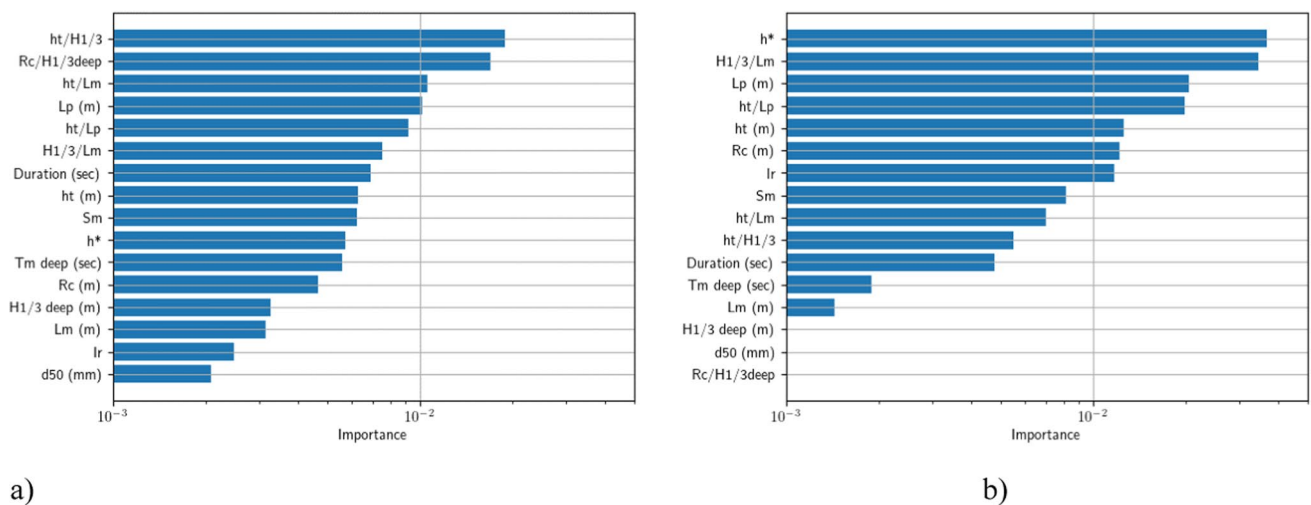


Fig. 6 Feature importance results for the (a) GBDT and (b) RF. A log scale is used on the horizontal axis that extended from 0.01 to 0.05

Table 5 Summary of the statistical performance comparison of ML models to predict scour depths

Model	Training dataset			Test dataset		
	RMSE	R^2 score	Pearson R	RMSE	R^2 score	Pearson R
GBDT	0.001	0.999	0.999	0.235	0.719	0.856
RF	0.003	0.999	0.999	0.230	0.729	0.858
SVR	0.147	0.905	0.956	0.310	0.509	0.715
RR	0.321	0.552	0.749	0.398	0.191	0.541

Summary and Conclusions

Given that the IPCC's future climate projections clearly indicate that climate change is an amplifier of current risks and pressures on coastal environments, it is critically important to ensure that the climate-informed coastal planning considers and understands the risks of failure of coastal sea defences and its consequences to the management of blue growth of coasts. With this context, the reliable predictions of scour depths of coastal protections, such as seawalls and breakwaters, are essential for the structural and functional safety of critical infrastructures in a changing climate. Therefore, recent advancements in applying data-driven techniques to predict coastal processes indicated the possibility of employing such robust algorithms in scour predictions. This study investigated four widely used ML algorithms (GBDT, SVR, RF, and RR) to predict scour depths at impermeable vertical seawalls with permeable shingle foreshores. The predictive models were trained and tested using measurements from small-scale 2D wave flume experiments on seawalls with two different sizes of permeable shingle slopes. Several model hyperparameters [GridSearchCV() and RandomizedSearchCV()] were configured and tested for each predictive ML model to investigate model overfitting and underfitting. The performance of each scenario was evaluated to identify the best model hyper-tuning configuration to estimate scour depths at seawalls, which improved the model performance by mitigating model overfitting and underfitting. Thus, an overall improvement in model performance was observed for the tested configurations. This improvement included the RandomizedSearchCV() method as the hyperparameter tuner compared to the GridSearchCV() method.

The resulting scour depths using the ML algorithms were compared against the experimental dataset of Salauddin et al. (Salauddin & Pearson 2019a). Overall, the predicted model values correlated well with the measured values, thus indicating the capability of the developed ML algorithms to predict scour depths at seawalls. Several statistical metrics, including the

R^2 score, RMSE, and Pearson R -value, were utilised to assess the statistical accuracy of the ML models in predicting scour depths. Compared to the statistical performance of employed ML algorithms, the overall RF and GBDT algorithms performed reasonably well. Nevertheless, the RF model slightly outperformed the GBDT with an RMSE value of 0.230, an R^2 value of 0.729, and a Pearson R -value of 0.858. Considering the marginal difference between the performance of RF and GBDT models, either ML approach would be an appropriate method to investigate scouring at vertical seawalls.

The RR algorithm (linear modelling approach) underperformed for the remaining models tested within this study. Nonetheless, this was expected as the RR algorithm retained the form of linear regression despite including a term for regularisation. Therefore, the findings of this study supported the assumption that linear models could not predict complex coastal processes, such as scouring at seawalls. These findings highlighted that the more sophisticated ML methods (GBDT, SVR, and RF) were more suitable for such investigations than traditional approaches. The study datasets consisted of 150 samples for the training and testing of ML models. Using a training-test split ratio of 70:30 implied that 105 and 45 samples were grouped into the training and test sets, respectively. Hence, each algorithm produced a relatively small training set to train the predictive model, which could lead to forming certain trends in model predictions. Further studies would be desirable with a relatively large dataset to mitigate the uncertainties associated with data size.

This study demonstrated that data-driven approaches, such as advanced ML algorithms (RF and GBDT), could be applied confidently in predicting scour depths at vertical seawalls with permeable shingle foreshores. The study findings should be used with existing scouring prediction methods to mitigate the damaging effects of scouring at vertical breakwaters with shingle slopes in a changing climate. Although ML methods, such as ANNs, were out of the study scope, this method still reported partial success in predicting coastal processes (overtopping at seawalls). Therefore, further research should be performed to compare the ANNs method with the current study algorithms.

Funding Open Access funding provided by the IReL Consortium

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abolfathi S, Yeganeh-Bakhtiary A, Hamze-Ziabari SM, Borzooei S (2016) Wave runup prediction using M5' model tree algorithm. *Ocean Eng* 112:76–81. <https://doi.org/10.1016/j.oceaneng.2015.12.016>
- Anscombe, F. J., 1973. Graphs in statistical analysis. *The American Statistician* 27(1), 17–21. <https://www.tandfonline.com/doi/abs/10.1080/00031305.1973.10478966>
- Bishop CM (2006) *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer Verlag, Berlin, Heidelberg
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B. & Varoquaux, G., 2013. API design for machine learning software: Experiences from the scikit-learn project. *API Design for Machine Learning Software: Experiences from the Scikit-learn Project*.
- den Bieman JP, van Gent MRA, van den Boogaard HFP (2021a). Wave overtopping predictions using an advanced machine learning technique. *Coastal Engineering*, 166. <https://doi.org/10.1016/j.coastaleng.2020.103830>
- den Bieman JP, van Gent MRA, van den Boogaard HFP (2021b) Wave overtopping predictions using an advanced machine learning technique. *Coastal Engineering*, 166. <https://doi.org/10.1016/j.coastaleng.2020.103830>
- Dong S, Abolfathi S, Salauddin M, Tan ZH, Pearson JM (2020a) Enhancing Climate Resilience of Vertical Seawall with Retrofitting - A Physical Modelling Study. *Applied Ocean Research* 103:102331. <https://doi.org/10.1016/j.apor.2020.102331>
- Dong S, Abolfathi S, Salauddin M, Pearson JM (2021) Spatial distribution of wave-by-wave overtopping behind vertical seawall with recurve retrofitting. *Ocean Engineering* 238:109674. <https://doi.org/10.1016/j.oceaneng.2021.109674>
- Dong S, Abolfathi S, Salauddin M, and Pearson JM (2020b) Spatial distribution of wave-by-wave overtopping at vertical seawalls. *Coastal Engineering Proceedings (36v)*. <https://doi.org/10.9753/icce.v36v.structures.17>
- Fowler JE (1992) Scour problems and methods for prediction of maximum scour at vertical seawalls. In: *Us Army Corps of Engineers, W. E. S. (eds.), Technical Report CERC-92-16, Coastal Engineering Research Center, Vicksburg, MS, USA.*
- Gíslason K, Fredsøe J, Deigaard R, Sumer BM (2009) Flow under standing waves. *Coast Eng* 56(3):341–362. <https://doi.org/10.1016/j.coastaleng.2008.11.001>
- Habib MA, O'Sullivan J, Salauddin M (2022a) Prediction of wave overtopping characteristics at coastal flood defences using machine learning algorithms: A systematic review. *IOP Conf. Ser.: Earth Environ. Sci.* 1072 012003. <https://iopscience.iop.org/article/10.1088/1755-1315/1072/1/012003/meta>
- Habib MA, O'Sullivan J, Salauddin M (2022b) Comparison of machine learning algorithms in predicting wave overtopping discharges at vertical breakwaters, *EGU General Assembly 2022, Vienna, Austria, 23–27 May 2022, EGU22-329*, <https://doi.org/10.5194/egusphere-egu22-329>
- Hughes SA (1996) *Physical Models and Laboratory Techniques in Coastal Engineering (Vol. 7)*. World Scientific. <https://doi.org/10.1142/2154>
- Jayaratne R, Edgar M, Rodolfo S, Garcia G, Francisco., (2015) *Laboratory Modelling of Scour on Seawalls*. In *Conf of Coastal Structures*, Boston
- Kuhn M, Johnson K (2013) *Applied Predictive Modelling*. Springer Science & Business Media
- Müller G, Allsop W, Bruce T, Kortenhaus A, Pearce A, Sutherland J (2008) The occurrence and effects of wave impacts. In: *Proceedings of the ICE-Maritime Engineering (ICE)*, pp. 167–173
- O'Sullivan JJ, Salauddin M, Abolfathi S, and Pearson JM (2020) Effectiveness of eco-retrofits in reducing wave overtopping on seawalls. *Coastal Engineering Proceedings (36v)*. <https://doi.org/10.9753/icce.v36v.structures.13>
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Müller A, Nothman J, Louppe G, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, & Duchesnay É (2011) Scikit-learn: Machine Learning in Python. <https://doi.org/10.48550/arXiv.1201.0490>
- Pourzangbar A, Losada MA, Saber A, Ahari LR, Larroudé P, Vaezi M, Brocchini M (2017a) Prediction of non-breaking wave induced scour depth at the trunk section of breakwaters using Genetic Programming and Artificial Neural Networks. *Coast Eng* 121:107–118. <https://doi.org/10.1016/j.coastaleng.2016.12.008>
- Pourzangbar A, Brocchini M, Saber A, Mahjoobi J, Mirzaaghasi M, Barzegar M (2017b) Prediction of scour depth at breakwaters due to non-breaking waves using machine learning approaches. *Appl Ocean Res* 63:120–128. <https://doi.org/10.1016/j.apor.2017.01.012>
- Powell KA, Lowe JP (1994) The scouring of sediments at the toe of seawalls. In: *Proceedings of the Hornafjordor International Coastal Symposium, Iceland*, pp. 749–755.
- Powell KA (1990) Predicting short term profile response for shingle beaches. Report SR 219, HR Wallingford, Wallingford, UK.
- Salauddin M, Pearson JM (2019a) Wave overtopping and toe scouring at a plain vertical seawall with shingle foreshore: A Physical model study. *Ocean Eng* 171:286–299. <https://doi.org/10.1016/j.oceaneng.2018.11.011>
- Salauddin M, Pearson JM (2019b) Experimental study on toe scouring at sloping walls with gravel foreshores. *J Mar Sci Eng* 7:198. <https://doi.org/10.3390/jmse7070198>
- Salauddin M, O'Sullivan JJ, Abolfathi S, Pearson JM (2021a) Eco-Engineering of Seawalls—An Opportunity for Enhanced Climate Resilience From Increased Topographic Complexity. *Frontiers of Marine Science* 8:674630. <https://doi.org/10.3389/fmars.2021.674630>
- Salauddin M, Pearson JM (2018) A laboratory study on wave overtopping at vertical seawalls with a shingle foreshore. *Coastal Engineering Proceedings*, 1(36), waves.56. <https://doi.org/10.9753/icce.v36.waves.56>

- Salauddin M, O'Sullivan JJ, Abolfathi S, Dong S, and Pearson JM (2020) Distribution of individual wave overtopping volumes on a sloping structure with a permeable foreshore. *Coastal Engineering Proceedings* (36v). <https://doi.org/10.9753/icce.v36v.papers.54>
- Salauddin M, Peng Z, Pearson J (2021b) The effects of wave impacts on toe scouring and overtopping concurrently for permeable shingle foreshores. EGU General Assembly 2021, online, 19–30 Apr 2021, EGU21-548. <https://doi.org/10.5194/egusp-here-egu21-548>
- Shetye A (2019) Feature selection with sklearn and pandas. URL: <https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b>. Accessed on 20th October 2022.
- Sumer BM, Fredsøe J (2000) Experimental study of 2D scour and its protection at a rubble-mound breakwater. *Coast Eng* 40:59–87
- Sutherland J, Brampton AH, Motyka G, Blanco B, Whitehouse RJW (2003) Beach lowering in front of coastal structures-Research Scoping Study. Report FD1916/TR, London, UK.
- Sutherland J, Obhrai C, Whitehouse R, Pearce A (2006) Laboratory tests of scour at a seawall. In: *Proceedings of the 3rd International Conference on Scour and Erosion, CURNET*, Technical University of Denmark, Gouda, The Netherlands
- Sutherland J, Brampton AH, Obrai C, Dunn S, Whitehouse RJW (2008) Understanding the lowering of beaches in front of coastal defence structures, Stage 2-Research Scoping Study. Report FD1927/TR, London, UK
- Sutton CD (2005) Classification and Regression Trees, Bagging, and Boosting, pp. 303–329. [https://doi.org/10.1016/S0169-7161\(04\)24011-1](https://doi.org/10.1016/S0169-7161(04)24011-1)
- Wallis M, Whitehouse R, Lyness N (2009) Development of guidance for the management of the toe of coastal defence structures. Presented in the 44th Defra Flood and Coastal Management Conference, Telford, UK
- Wolpert D, Macready W (1996) No free lunch theorems for search. SFI WORKING PAPER: 1995–02–010. Santa Fe Institute
- Xie SL (1981) Scouring patterns in front of vertical breakwaters and their influences on the stability of the foundation of the breakwaters. Delft University of Technology, Delft, The Netherlands, Department of Civil Engineering

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.