Wave overtopping predictions using an advanced machine learning technique

den Bieman, Joost P.; van Gent, Marcel R.A.; van den Boogaard, Henk F.P.

DOI
10.1016/j.coastaleng.2020.103830

Publication date
2021

Document Version
Accepted author manuscript

Published in
Coastal Engineering

Citation (APA)
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*, 1-12. Article 103830. https://doi.org/10.1016/j.coastaleng.2020.103830

Wave overtopping predictions using an advanced machine learning technique

Joost P. den Bieman[a,*], Marcel R. A. van Gent[a,b], Henk F. P. van den Boogaard[a]

[a] *Deltares, Department of Coastal Structures and Waves (JdB & MvG) & Deltares Software Centre (HvdB), Boussinesqweg 1, 2629HV Delft, The Netherlands.*
[b] *TU Delft, Department of Hydraulic Engineering, Stevinweg 1, 2628CN Delft, The Netherlands.*

Abstract

Coastal structures are often designed to a maximum allowable wave overtopping discharge, hence accurate prediction of the amount of wave overtopping is an important issue. Both empirical formulae and neural networks are among the commonly used prediction tools. In this work, a new model for the prediction of mean wave overtopping discharge is presented using the innovative machine learning technique XGBoost. The selection of features to train the model on is carefully substantiated, including the redefinition of existing features to obtain a better model performance. Confidence intervals are derived by tuning hyperparameters and applying bootstrap resampling. The quality of the model is tested against four new physical model data sets, and a thorough quantitative comparison with existing machine learning methods and empirical overtopping formulae is presented. The XGBoost model generally outperforms other methods for the test data sets with normally incident waves. All data-driven methods show less accuracy on oblique wave data, presumably because these conditions are underrepresented in the training data. The performance of the XGBoost model is significantly improved by adding a randomly selected part of the new oblique wave cases to the training data. In the end, this new model is shown to reduce errors on all data used in this work with a factor of up to 5 compared to existing overtopping prediction methods.

Keywords: Machine learning; Wave overtopping; Coastal structures; Physical

*Corresponding author

Email addresses: `joost.denbieman@deltares.nl` (Joost P. den Bieman), `marcel.vangent@deltares.nl` (Marcel R. A. van Gent), `henk.vandenboogaard@deltares.nl` (Henk F. P. van den Boogaard)

model tests; Gradient boosting decision trees; XGBoost

1. Introduction

Wave overtopping has the potential to interfere with the function of a coastal structure and cause structural damage or physical harm. To reduce these risks, coastal structures are often designed to prevent exceeding a maximum allowable wave overtopping discharge. Therefore, estimates of the amount of wave overtopping are important for the design of coastal structures.

Currently, different types of tools are available to predict the expected amount of wave overtopping, given a certain configuration of a coastal structure. Firstly, many empirical overtopping formulae have been derived from physical model data. These form a relatively easy estimate of the mean wave overtopping discharge, q [$\mathrm{m}^3/\mathrm{s}/\mathrm{m}$]. A selection of those formulae are listed in TAW (2002) and in the EurOtop manual (EurOtop, 2018). The so-called CLASH database (Steendam et al., 2004) with wave overtopping data from measurements has been used by Van Gent et al. (2007) as training data for a neural network (NN) to predict wave overtopping. Their ensemble of NNs outputs both the expected mean wave overtopping discharge and an estimate for the corresponding uncertainty. A similar approach was used while extending both the training data set and adding predictions of wave transmission and reflection (Zanuttigh et al., 2016). Recently, it was shown in Den Bieman et al. (2020) that the machine learning method XGBoost (Chen & Guestrin, 2016) can be successfully applied as an alternative to NN models. XGBoost is a relatively new method, finding success in various practical applications from fault detection in wind turbines (Zhang et al., 2018) to bridge damage estimation (Lim & Chi, 2019). Applying the method to the prediction of wave overtopping significantly reduces the prediction errors on the CLASH database compared to the NN by Van Gent et al. (2007), see Den Bieman et al. (2020). In addition to empirical formulae and machine learning methods, numerical models are capable of reproducing physical wave overtopping models reasonably well. Hence, numerical modelling could also be used to predict mean wave overtopping discharge, on the condition that extensive calibration and validation on physical model data has been carried out.

The exploratory work in Den Bieman et al. (2020) compares the existing NN model by Van Gent et al. (2007) to an XGBoost model with a similar setup that is trained on the same training data set. The XGBoost method is shown to outperform the NN, reducing errors by a factor of 2.8. In this paper, that work

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

is expanded upon in several ways to get to a state-of-the art XGBoost model for the prediction of mean wave overtopping discharges. Firstly, the training database is enlarged beyond the original CLASH database and the selection of features for model training is carefully substantiated. Secondly, both the hyperparameter tuning and derivation of uncertainties is readdressed, as Den Bieman et al. (2020) find surprisingly small confidence intervals. Thirdly, the XGBoost model is validated on both the overtopping database and new physical model data previously unseen by the model. Finally, the model is compared with predictions from two of the available neural network models (Van Gent et al., 2007; Zanuttigh et al., 2016) and from two empirical overtopping formulae (TAW, 2002; EurOtop, 2018).

This article is structured as follows. Section 2 contains the description of the machine learning methods and the training and test data sets that have been used. Section 3 expands upon feature engineering, hyperparameter tuning, and uncertainty estimation. The model performance is quantified in Section 4, using both the overtopping database and the test data sets. In Section 5, a discussion of the results is presented. Finally, Section 6 contains conclusions and recommendations.

2. Method description

In the following, the methods used in this paper are expanded upon: the machine learning methods applied (Section 2.1), the data used to train them (Section 2.2), the new test data sets (Section 2.3), and the other overtopping prediction methods that are used for comparison (Section 2.4).

2.1. XGBoost and gradient boosting decision trees

XGBoost (Chen & Guestrin, 2016) is a Python (Van Rossum, 1995) implementation of a machine learning method of the type gradient boosting decision trees (GBDT). These methods are based on decision trees that can solve either classification problems (predicting a label) or regression problems (predicting a quantity). These decision trees are therefore often called classification and regression trees (CART). In this work regression trees are used for the prediction of mean wave overtopping discharges at coastal structures.

Figure 1 shows an ensemble of three decision trees, which each consist of decision and leaf nodes. In decision nodes (D_{ij}), a condition is defined based on a feature from the training data. This combination of feature and condition is often called a split. Node D_{11} for example could contain the condition: "Is

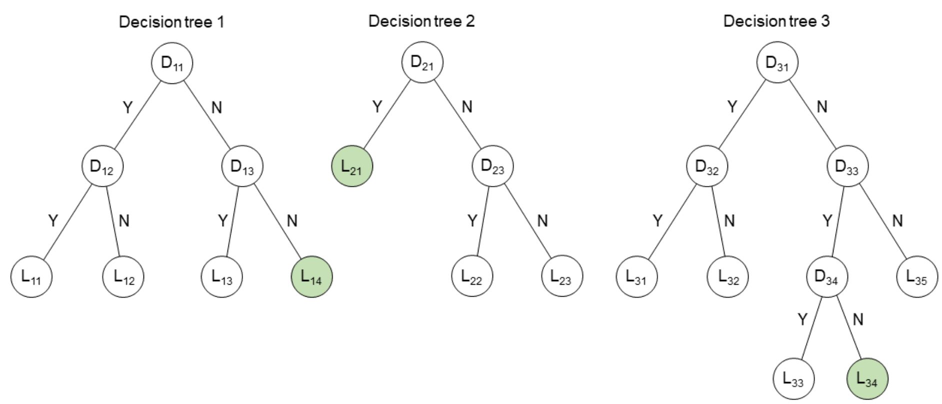


Figure 1: Schematic depiction of an ensemble of decision trees with desicion (D_{ij}) and leaf nodes (L_{ij}). An example prediction for one combination of input parameters is shown in green. Source: Den Bieman et al. (2020)

the berm width larger than 0 m?". Two tree branches emerge from the node, one for each possible answer (Yes or No) to the question. These feed into either another decision node or a leaf node. Leaf nodes (L_{ij}) form the end points of the tree and contain the prediction. Leaves of regression trees predict values, whereas leaves of classification trees predict classes. The depth of a decision tree is defined as the number of subsequent decision nodes from start to leaf (i.e. decision trees 1 and 2 depicted in Figure 1 have a depth of 2, while decision tree 3 has a depth of 3).

In practice, many classification or regression problems are far too complex to solve with a single decision tree. Hence, GBDT methods use a large amount of trees in an ensemble. The basic principle underlying an ensemble of decision trees is that a combination of weak predictors can form a strong predictor. The prediction of the ensemble is the sum of the predictions of the individual trees (see the green leaf nodes in Figure 1 for example), taking the learning rate into account (see Section 3.1). Newly added trees seek to correct the prediction errors of the existing trees within the ensemble. In this way, the prediction error is iteratively reduced. The total amount of trees in the ensemble can either be specified beforehand or determined on the fly based on the error reduction (often referred to as "early stopping"). The latter is applied in this work and is further explained in Section 3.3. When determining the configuration of a tree, its splits need to be determined. First an objective function is defined that both rewards accurate predictions and penalizes tree complexity. The algorithm starts at a tree depth of 0 and iteratively adds levels of tree depth. For every level, it finds the optimal condition and leaf values for the split per feature. Subsequently, the feature and split that result in the largest improvement of the objective function is used in the decision node, growing the tree one level deeper. The tree is grown up to the maximum tree depth. A more detailed description of the algorithm is given by Chen & Guestrin (2016).

The use of an ensemble of decision trees results in a flexible resolution, depending on the local density of training data. This is especially useful given the large density differences in overtopping databases. Note that, as a result, GBDT methods are generally expected to be less suitable for extrapolation far beyond the coverage of the training data.

2.2. Training data set

Currently, the available NN models are the model by Van Gent et al. (2007), hereafter also referred to as "NN", and the model by Zanuttigh et al. (2016), hereafter also referred to as "NNb". In this work, the XGB model performance is compared to both NN, NNb, and empirical overtopping formulae (TAW, 2002; EurOtop, 2018). The NN model is trained on a selection of entries from the original CLASH database (Steendam et al., 2004). The NNb model by Zanuttigh et al. (2016) uses an extended version of the CLASH database as training data set. The extended database adds additional data on vertical walls (Oumeraci et al., 2007), rubble mounds with cobs (Besley et al., 1993), reshaping berm breakwaters (Lykke Andersen et al., 2008), smooth steep slopes (Victor & Troch, 2012), and smooth slopes with walls (Van Doorslaer et al., 2015). This additional data has been merged with the CLASH database into the database used by Zanuttigh et al. (2016). This will be referred to as the "overtopping database" in the rest of this paper. The overtopping database has been randomly split 80%/20% into two parts: a "training data set" (6943 records) used for training the XGB model , and a "test data set" (1736 records) which is kept strictly separate and is only used to demonstrate the predictive quality of the final trained model. Finally, the new data (from four new data sets described in Section 2.3) is referred to as "additional test data sets" or "unseen data".

Not all available parameters from the overtopping database are used in model training. Those parameters that are used to train a model are called features. In Table 1 and Figure 2, the features used in the training of one or more models (NN, NNb and/or XGB) are respectively listed and illustrated. This includes the additions that follow from feature engineering, as described in Section 3.2. The target variable used in model training is the $\log_{10}$ of the mean wave overtopping discharge q after Froude scaling.

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

As in Van Gent et al. (2007), Froude's similarity law is used to scale the overtopping database features to $H_{m0,toe} = 1$ m, which is indicated in the right-most column of Table 1. This scaled data is used in the model training detailed in Section 3. After being used for scaling, the feature $H_{m0,toe}$ is no longer used in model training. Similarly, the complexity (CF) and reliability factors (RF) are not directly used for model training. They serve strictly for the weighting of the training data records. Both factors take on integer values of 1 (low complexity, high reliability) through 4 (high complexity, low reliability). The weight factor (WF) is determined with the formula from Van Gent et al. (2007): $WF = (4 - RF) \cdot (4 - CF)$. This formula gives the highest WF to the most reliable and least complex training data. Very unreliable ($RF = 4$) or complex ($CF = 4$) are excluded from the training data. In the end this results in a total of 8679 records in the overtopping database.

Additionally, Van Gent et al. (2007) state that measurements of very small mean wave overtopping discharges can be strongly affected by scale effects, and thus are less reliable. The practical application or relevance of discharges smaller than 0.001 l/m/s is also quite low. Therefore they suggest applying $WF = 1$ to all entries with $q < 10^{-6}$ m^3/s/m (before Froude scaling) and disregarding their associated reliability and complexity factors. This suggestion is adopted and applies to 1060 of the 8679 records.

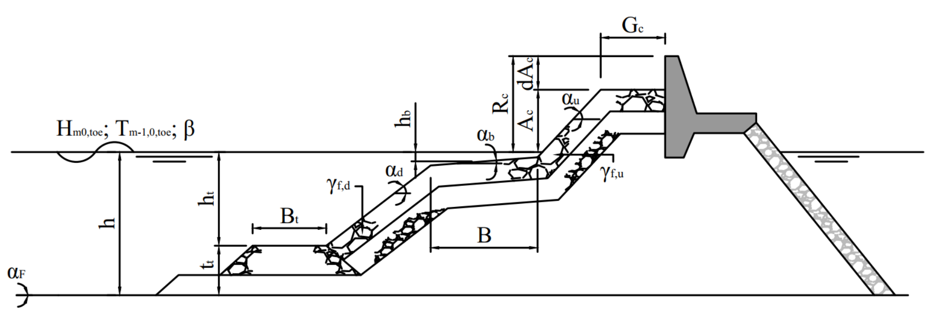


Figure 2: Feature definitions, adapted from Van Gent et al. (2007).

2.3. Additional test data sets

Next to the training data described in Section 2.2, several additional test data sets from recent physical model experiments are used to evaluate the model performance. These additional test data sets are as of yet unseen by any of the machine learning models, i.e. they are not part of the data the NN, NNb and

Table 1: Overview of features used in model training in the NN by Van Gent et al. (2007), the NNb by Zanuttigh et al. (2016) and the new XGB model.

Name	Symbol	Unit	NN	NNb	XGB	Fr scaled
Mean wave overtopping discharge	q	[m^3/s/m]	✓	✓	✓	✓
Water depth, toe	h	[m]	✓	✓	✓	✓
Spectral significant wave height, toe	$H_{m0,toe}$	[m]	✓	✓	✓	-
Spectral wave period, toe	$T_{m-1,0,toe}$	[s]	✓	✓	✓	✓
Angle of wave attack	β	[°]	✓	✓	✓	-
Roughness factor of the structure	γ_f	[-]	✓	✓	-	-
Roughness factor of the lower slope	$\gamma_{f,d}$	[-]	-	-	-	-
Roughness factor of the upper slope	$\gamma_{f,u}$	[-]	-	-	✓	-
Ratio of roughness factors	$f_{\gamma_f} = \frac{\gamma_{f,d}}{\gamma_{f,u}}$	[-]	-	-	✓	-
Cotangent of the lower slope	$\cot\alpha_d$	[-]	✓	✓	✓	-
Cotangent of the upper slope	$\cot\alpha_u$	[-]	✓	-	✓	-
Cotangent of the average slope	$\cot\alpha_{incl}$	[-]	-	✓	-	-
Crest freeboard	R_c	[m]	✓	✓	✓	✓
Armour crest freeboard	A_c	[m]	✓	✓	-	✓
Difference between crest and armour crest freeboard	$dA_c = A_c - R_c$	[m]	-	-	✓	✓
Crest width	G_c	[m]	✓	✓	✓	✓
Width of the berm	B	[m]	✓	✓	✓	✓
Water depth above the berm	h_b	[m]	✓	✓	✓	✓
Tangent of berm slope	$\tan\alpha_B$	[-]	✓	-	✓	-
Water depth above the toe structure	h_t	[m]	✓	✓	-	✓
Thickness of the toe structure	$t_t = h - h_t$	[m]	-	-	✓	✓
Width of the toe structure	B_t	[m]	✓	✓	✓	✓
Element size structure	D	[m]	-	✓	-	✓
Cotangent of foreshore slope	$m = \cot\alpha_F$	[-]	-	✓	-	-
Tangent of foreshore slope	$\tan\alpha_F$	[-]	-	-	✓	-
Complexity factor	CF	[-]	✓	✓	✓	-
Reliability factor	RF	[-]	✓	✓	✓	-

XGB models are trained on. In Table 2, the relevant parameter ranges covered by these additional test data sets are listed. The individual data sets are described in more detail below. Part of these data sets contain situations that are underrepresented in the overtopping database; i.e. these data sets contain elements in regions with little or no coverage, or may be situated in remote corners of the database. For such data records it can be difficult for data-driven models to obtain accurate and meaningful predictions.

Table 2: Overview of the relevant parameter ranges covered by the additional test data sets and the examples shown in Figure 3.

Symbol	DS 1a	DS 1b	DS 2	DS 3	DS 4	Fig. 3
$R_c/H_{m0,toe}$	0.9 - 1.8	0.9 - 2.1	0.8 - 2.1	0.8 - 2.2	1.0 - 3.0	1 - 1.5
$dA_c/H_{m0,toe}$	0	0	-0.7 - 0	-0.8 - -0.2	0	-0.5 - 0
$B/H_{m0,toe}$	0 - 2.2	1.4 - 2.2	0	0	1.1 - 2.5	0.5
$h_b/H_{m0,toe}$	-0.3 - 0.3	-0.5 - 0.5	0	0	-0.42 - 0.42	0
$\cot\alpha_d$	3	3	2	2	3	4
$\cot\alpha_u$	3	3	2	2	3	2.5
$\gamma_{f,u}$	0.4 - 1.0	0.5 - 1.0	0.4	0.45	0.8	0.5
f_{γ_f}	1.0	0.5 - 2.0	1.0	1.0	1.0	1.0
$h/H_{m0,toe}$	4.3 - 6.7	4.3 - 6.5	4.1 - 10.2	3.5 - 11.6	2.7 - 6.7	1.5
$s_{m-1,0}$ [%]	2.7 - 4.9	1.3 - 4.2	1.3 - 4.2	1.4 - 4.8	1.7 - 4.2	2.4
β [°]	0	0	0	0 - 75	0 - 75	0
$\tan\alpha_F$	0	0	0	0	0	0

Data Set 1 (362 records) comes from physical model studies of the influence of roughness on wave overtopping at dikes and rock structures (Chen et al., 2020a,b). These experiments feature different revetment types, including roughness differences between upper and lower slope. None of the existing overtopping prediction methods properly take those roughness differences into account, except for the method proposed by Chen et al. (2020b). As a consequence, the prediction methods applied here are expected to be less accurate for the entries with roughness differences than they will be for entries with constant roughness. Hence, in the following the data set will be split into two parts: Data Set 1a (206 records) only contains data with constant roughness, while Data Set 1b (156 records) exclusively contains the data records with roughness differences between the upper and lower slopes.

Data Set 2 (51 records) contains physical model experiment data of a rock structure with a crest wall (Jacobsen et al., 2018).

Data Set 3 (242 records) features physical model experiment data of wave overtopping on rubble mound breakwaters with a crest wall under oblique wave attack (Van Gent & Van der Werf, 2019).

Data Set 4 (177 records) consists of data from physical model experiments of impermeable slopes with a berm under oblique wave attack (Van Gent, 2020).

In general, Data Sets 1a and 2 are expected to be within the range of the data already present in the training data set. Data Set 1b features roughness differences between the lower and upper slope, which is relatively rare in the training data (2.8% of all records). Data Sets 3 and 4 feature oblique wave attack which is also underrepresented in the training data (10.9% of entries), especially when combined with the presence of a crest wall (1.1% of entries) or a berm (1.0% of entries). Note that the constructions used in Data Set 1-4 are not complex and can be described exactly following the hydraulic structure definitions in Figure 2. Hence $CF = 1$ for all above-mentioned additional test data sets.

2.4. Other overtopping prediction methods

The XGB model results and the performance on measurement data are compared to those of other often used overtopping prediction methods. The methods compared to in this work are the empirical formulae from TAW (2002), the second edition of the EurOtop manual (EurOtop, 2018), the NN by Van Gent et al. (2007), and the NNb by Zanuttigh et al. (2016).

The TAW and EurOtop manuals contains a selection of empirical formulae that predict mean wave overtopping discharge. Two versions of these formulae are presented; a mean value approach that represents a best fit with data, and a design and assessment approach which includes some conservatism. In this work, the best fit with data is of importance, hence only results from the mean value approach are considered.

Van Gent et al. (2007) made use of machine learning methods by applying a NN to predict mean wave overtopping discharge. They use an ensemble of NNs that gives both the expected discharge and the associated prediction uncertainty as an output. Their NN is available through the NN-Overtopping web application (Deltares). Zanuttigh et al. (2016) continues on the same concept but makes use of a combination of a classifier model coupled to three separate neural networks. They used slightly different features to describe the characteristics of the hydraulic structure (see Table 1).

3. Model training and tuning

Training and tuning a machine learning model comprises of several different steps. Section 3.1 describes the process of tuning the different hyperparameters of the XGB model. In Section 3.2, several features from the overtopping database are redefined to be more suitable for use in machine learning methods. Additionally, the process of coming to a final selection of features to train the model on is explained. Section 3.3 deals with the derivation of confidence intervals using bootstrap resampling.

3.1. Hyperparameter tuning

The term hyperparameter refers to the run control parameters of a machine learning method. For XGB, these run control parameters govern the complexity and architecture of individual decision trees. Without any restriction to complexity, the model is expected to be overfitted on the training data, losing any generic predictive skill.

The XGB hyperparameters that have been tuned are listed in Table 3 and can be explained as follows. The maximum depth of a single decision tree (max_depth) restricts the number of subsequent splits in decision nodes. The values used in the tuning process are chosen to stay within the total number of features in the training data set. Furthermore, when growing the tree each leaf node must contain a minimum number of data points (min_child_weight). In this case, leaf nodes with a single data point are not allowed and up to twelve are required. Leaf node values are multiplied by a learning rate $learning\_rate$ to obtain a slower convergence that reduces overfitting. Learning rates of < 0.1 are very common in machine learning, to which the chosen values in Table 3 adhere. Note that both the complexity regularization (reg_lambda) and the subsampling ($subsample$) terms are not included in hyperparameter tuning. The reason for excluding these parameters is that it leads to more realistic uncertainty estimates, as is further described in Section 3.3. Both hyperparameters are set to their default values, with mild regularization ($reg\_lambda = 1$) and no subsampling ($subsample = 1$).

The optimal hyperparameter values are listed in Table 3 (indicated in blue). These optimal values are found with a K-fold cross-validation (with $K = 5$) combined with a grid search. In the K-fold cross-validation, the total data set is split into K parts (or folds). One of the folds is used for model validation, while the rest is used for model training. This is repeated K times, with a different fold used for validation each time. Hence, the choice of $K = 5$ uses 80% of the data for training, the remaining 20% for validation, and is repeated 5 times with a different part used for validation. The K-fold cross-validation is performed in a grid search, which uses every combination of parameters listed in Table 3. Finally, the best performing hyperparameter set is selected. This hyperparameter set prescribes rather shallow trees, with a reasonable amount of data points per leaf and a rather large learning rate.

Table 3: XGB parameter combinations used in the K-fold cross-validation, optimal values in blue.

Name	Parameter name	Values
Max. tree depth	*max_depth*	6; 7; 8; 10; 12; 14
Min. data points per leaf	*min_child_weight*	3; 5; 7; 10; 12
Learning rate	*learning_rate*	0.005; 0.0075; 0.01; 0.02; 0.05

3.2. Feature engineering and feature importance

Feature engineering is a common step in the process of improving machine learning models. The parameters selected from the data to train a model on are called features. Feature engineering as a term refers to deriving new features that add to or replace existing parameters in the training data set. The intent of feature engineering is to derive new features that provide a better description of the dependencies and sensitivities of the target variable ($\log_{10}(q)$ in this case) to the model input. Note that successful feature engineering depends heavily on the characteristics of the data set in question. Hence, there is no single approach that always leads to good results.

One often used approach in machine learning is to perform a feature importance analysis. This type of analysis seeks to quantify the influence of individual features on the target variable, which is useful information in the decision to in- or exclude features in model training. A permutation importance analysis (Breiman, 2001; Fisher et al., 2018) is performed to gain insight into the influence of each feature. In this method, the data is split into a test and a training data set, the latter of which is used to train a single model. For one feature at a time, the test data set values are randomly scrambled. Subsequently, the trained model is used to generate predictions for the test data set. For important features, the scrambling should have a large effect on the prediction of q compared to the unscrambled test data set, whereas the influence will be small for unimportant features. All features are scrambled one-by-one, with the scrambling repeated 5 times to account for the effect of random sampling. The ELI5 (ELI5) Python permutation importance implementation has been used in this paper. In Table 4, the weight and standard deviation (σ) resulting from the permutation analysis are listed for both a selection of features similar to the NNb model (using Froude scaled features and replacing both A_c and h_t with dA_c and t_t respectively) and the candidate features for the XGB model.

Using uncorrelated features is imperative to obtain an accurate representation of the importance of each feature. An accurate overview of which features are (un)important to predict the mean wave overtopping discharge enables a well-argued choice for the selection of features used in a machine learning method. Simply using all features unnecessarily increases the computational demand of model training, can promote overfitting, and can reduce the generic applicability of the model. The results of the permutation importance analysis listed in Table 4 support redefinition or removal of certain highly correlated features, which is explained below.

Added information in the overtopping database allows for distinguishing differences between the roughness of the upper ($\gamma_{f,u}$) and lower slope ($\gamma_{f,d}$) of the structure. Since there are many entries in the database with the same roughness on both slopes, $\gamma_{f,u}$ and $\gamma_{f,d}$ will be highly correlated in practice. Since uncorrelated features are preferred, the XGB model uses the ratio between lower and upper slope roughness $f_{\gamma_f} = \frac{\gamma_{f,d}}{\gamma_{f,u}}$ instead of $\gamma_{f,d}$. Similarly, two additional features are made uncorrelated. Firstly, the armour crest freeboard (A_c) is made uncorrelated from the crest freeboard (R_c) by using the difference between both as a feature: $dA_c = A_c - R_c$. Secondly, the water depth above the toe structure (h_t) is replaced by the thickness of the toe structure ($t_t = h - h_t$) to remove the correlation with the water depth (h).

In contrast to the NN, the NNb also uses the element size of the structure (D) as a feature. Zanuttigh et al. (2016) indicate that a weighted average of the element size in the wave run-up and run-down area is taken as the representative element size. Next to the mean wave overtopping discharge, the NNb is also used to predict wave reflection and wave transmission coefficients for the given structure, for which the element size is of significant importance. In the context of wave overtopping however, D relates in large part to the roughness of the profile, which is already represented by $\gamma_{f,u}$ and either $\gamma_{f,d}$ or f_{γ_f}. Table 4 also illustrates that the importance of γ_f (left-hand side) is significantly smaller than that of $\gamma_{f,u}$ (right-hand side), since the roughness in the NNb is represented by both γ_f and D. Thus, since the main effects of the element size are already present in the parameter(s) accounting for the roughness, there seems to be no benefit to including D as a feature in wave overtopping prediction models. Analogously, the importance of the berm width (B) is also diminished (left-hand side) when it is already implicitly included in the average slope ($\cot\alpha_{incl}$). Replacing the average slope with the upper slope ($\cot\alpha_u$) resolves the problem.

In addition to reducing the amount of highly correlated features, the cotangent of the foreshore slope m is replaced with the tangent of the foreshore slope, $\tan\alpha_F$. In this way, $\tan\alpha_F$ will be 0 for cases without a foreshore.

Table 4: Overview of permutation importance of XGB models with NNb-like feature set and overview of candidates. Features selected in the XGB model are indicated by (✓).

	NNb features		Overview of candidates	
Rank	Feature	Weight $\pm\,\sigma$	Feature (selected)	Weight $\pm\,\sigma$
1	R_c	0.9178 ± 0.0405	R_c (✓)	0.8899 ± 0.0279
2	γ_f	0.3093 ± 0.0124	$\gamma_{f,u}$ (✓)	0.4610 ± 0.0197
3	$T_{m-1,0,toe}$	0.1702 ± 0.0041	G_c (✓)	0.1922 ± 0.0081
4	G_c	0.1632 ± 0.0106	$T_{m-1,0,toe}$ (✓)	0.1530 ± 0.0066
5	$\cot\alpha_{incl}$	0.0824 ± 0.0047	B (✓)	0.0799 ± 0.0030
6	D	0.0810 ± 0.0045	$\cot\alpha_u$ (✓)	0.0641 ± 0.0025
7	β	0.0460 ± 0.0021	β (✓)	0.0433 ± 0.0021
8	h	0.0411 ± 0.0037	h (✓)	0.0417 ± 0.0030
9	B	0.0378 ± 0.0025	$\tan\alpha_F$ (✓)	0.0405 ± 0.0017
10	m	0.0364 ± 0.0024	$\cot\alpha_d$ (✓)	0.0236 ± 0.0028
11	$\cot\alpha_d$	0.0194 ± 0.0019	dA_c (✓)	0.0234 ± 0.0020
12	dA_c	0.0150 ± 0.0011	t_t (✓)	0.0200 ± 0.0024
13	t_t	0.0150 ± 0.0027	h_b (✓)	0.0154 ± 0.0016
14	h_b	0.0143 ± 0.0006	B_t (✓)	0.0093 ± 0.0012
15	B_t	0.0063 ± 0.0019	f_{γ_f} (✓)	0.0002 ± 0.0001
16	-	-	$\tan\alpha_B$ (-)	0.0000 ± 0.0001

The selection of features for the XGB model is indicated with check marks in Table 4. The berm slope ($\tan\alpha_B$) is not used to train the model, since it ranks as the least important feature and its importance in absolute terms is very small. Surprisingly, the newly introduced feature f_{γ_f} also ranks low on importance, while Chen et al. (2020b) show the importance of taking into account roughness differences. One of the likely causes for this discrepancy is that only 2.8%

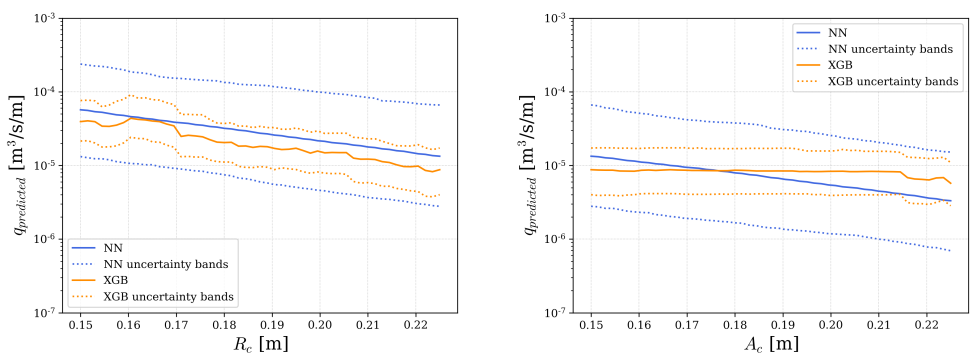
of the entries in the overtopping database has a difference between the roughness on the upper and lower slopes, and conversely $f_{\gamma_f} = 1.0$ for 97.2% of the data. This means that scrambling the feature in the permutation importance analysis will give the same value for f_{γ_f} in many cases, and thus a seemingly small importance is attributed to the feature. Hence, f_{γ_f} is still included as a feature in model training.

3.3. Bootstrap resampling and confidence intervals

For the sake of consistency and comparability of the results, the bootstrap resampling method (Efron & Tibshirani, 1993) - proposed by Van Gent et al. (2007) and also used by Zanuttigh et al. (2016) - is similarly applied to the XGB model to obtain estimates of prediction errors. Note that there might be other suitable methods for the estimation of predictions errors, but these are not explored in this work. The bootstrap resampling method can be summarized as follows. Firstly, 500 bootstrap resamples are generated from all data available for model training. A resample is a randomized selection from the overtopping database, where individual entries can be selected more than once. When that is the case, the weight factor for that entry is adjusted accordingly within the resample. Subsequently, a model is trained for each resample. The resample is used as a training data set. The training makes use of an "early stopping" algorithm. This algorithm keeps adding new trees to the model, until either the maximum number of trees (set to 100.000) is reached or if the last 1000 consecutively added trees do not improve the model prediction for the entries not selected in the bootstrap resample. In the latter case, the model training is stopped and the best model is selected as the training result. In this way, 500 models are trained with 500 different but overlapping data sets and no individual model is trained on all available data from the overtopping database. Finally, for each prediction all 500 models are used, from which the median value serves as model prediction and the associated error can be estimated.

The use of specific anti-overfitting parameters in XGB tends to generalize the model fits to such degree that the variation in the model predictions - and thus the estimated confidence intervals - is greatly reduced. Hence, in the hyperparameter tuning for this work (Section 3.1), subsampling is not applied and a milder tree complexity regularization is used. In Figure 3, predictions of the current XGB model and their associated 90% confidence interval are compared to the NN model for singular variations along both crest and armour crest freeboard, with constant values for other features. The parameter ranges for these laboratory scale examples are listed in Table 2. As can be seen in Figure 3a and

Figure 3b, the updated newly tuned hyperparameter settings lead to seemingly realistic uncertainty bands. The prediction uncertainty bands are expected to be smaller that those of the NN model, since the XGB model performance is generally better (see Section 4). Note that the NN model generally leads to a smoother trend than the XGB model. This is an inherent feature of the decision tree based machine learning method applied here. The many splits in an ensemble of decision trees inadvertently introduce some amount of discontinuity to the predicted overtopping discharge over a given parameter range. Hence, it is not related to the approach used to derive uncertainty estimates. Additionally, further analysis with 1000 resamples shows no significant differences, which suggests that the 90% interval can be adequately determined from 500 resamples.



(a) Model responses to a variation in R_c, with a constant $A_c/H_{m0,toe} = 1.0$.

(b) Model responses to a variation in A_c, with a constant $R_c/H_{m0,toe} = 1.5$.

Figure 3: Examples of predictions and 90% uncertainty bands for NN (blue) and XGB (orange) models for a changing crest and armour crest freeboard (parameter ranges listed in Table 2).

4. Model validation

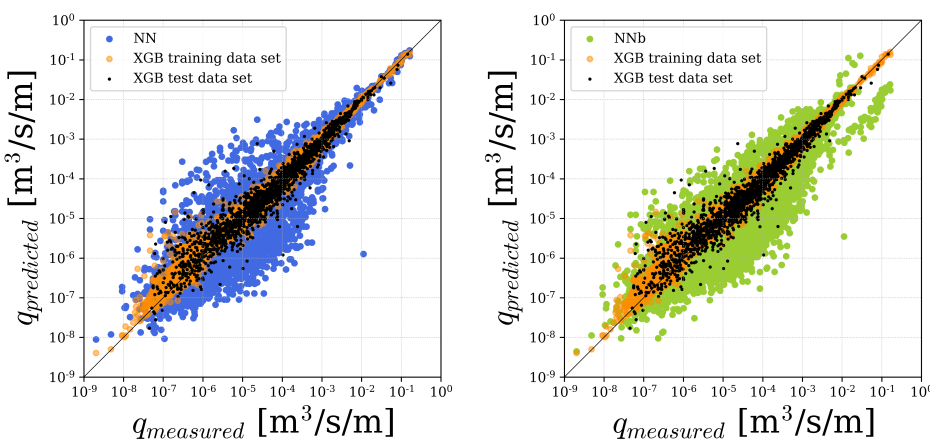
Validation of the XGB model is performed in several steps. Firstly, in Section 4.1 its performance is analyzed on the overtopping database and its predictive skill is demonstrated using the test data set. Subsequently, in Section 4.2 the generic applicability of the model is analyzed by applying the model to the challenging conditions of the additional test data, which was not used in any way in the model training. Finally, the model validation after retraining with the expanded training data set (indicated by "XGBr") is considered in Section 4.3.

Preprint, accepted for publication in Coastal Engineering: `https://doi.org/10.1016/j.coastaleng.2020.103830`

4.1. Validation on the overtopping database

The performance of the XGB model is evaluated on the test data set (see Section 2.2). Additionally, the prediction errors are compared to those of the other existing tools to predict overtopping discharges. This primarily concerns the original NN model (Van Gent et al., 2007), but also includes the NNb model (Zanuttigh et al., 2016) and the empirical overtopping formulae (TAW, 2002; EurOtop, 2018) for wave overtopping prediction. The weighted root-mean-square-error (RMSE) is used as an error criterion. It is defined by Equation 1 and listed for all methods in Table 5.

$$RMSE = \sqrt{\frac{1}{\sum_{n=1}^{N}(WF_n)}\frac{1}{N}\sum_{n=1}^{N}(WF_n \cdot \left(\log_{10}(q_{predicted,n}) - log_{10}(q_{measured,n})\right)^2)} \quad (1)$$



(a) NN and XGB.

(b) NNb and XGB.

Figure 4: Predictions by the NN (blue) and NNb (green) models for the overtopping database, and by the XGB model for the training data set (orange) and the test data set (black).

The predictions of the different machine learning methods for the overtopping database are shown in the scatter plots of Figure 4. The larger RMSEs of the NN and NNb models translate into a large scatter around the diagonal, with prediction errors of up to a factor 100. The scatter in the XGB model predictions is visibly much smaller, with differences largely within a factor 10. This is reflected in Table 5, where a significantly smaller test data set RMSE is listed for

XGB (0.284) than for NN (0.478) and NNb (0.580). The same holds for the training data set and the entire overtopping database. Note that the entire overtopping database was used as a training data set for the NNb model, while a smaller subset thereof was available to train the NN model.

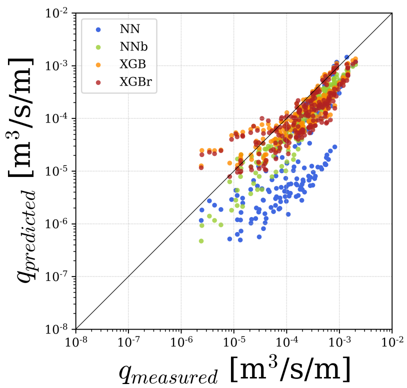
4.2. Validation on additional test data sets

In Section 4.1 the capability of the XGB model to predict the contents of the test data set is shown. The good performance on the test data set shows that the XGB model has predictive capabilities on previously unseen data that is fairly similar to the training data. It does not, however, show the predictive capability of the model for conditions that are meaningfully different from the training data set (in this case the overtopping database). As mentioned in Section 2.3, conditions that are not in the overtopping database but are very similar to it, should be predicted reasonably well. The real challenge lies in conditions that are not well represented in or covered by the overtopping database. For instance, conditions with oblique wave attack are relatively sparse in the overtopping database. Hence, there is relatively little data available for the model to correctly learn and predict wave overtopping discharges under oblique wave attack.

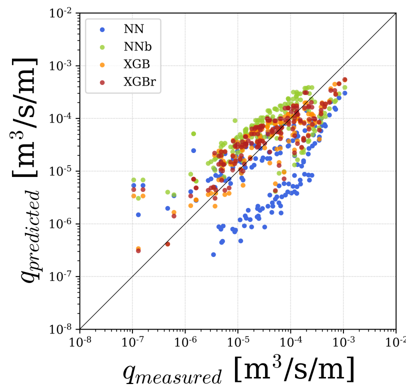
Table 5: RMSE for all overtopping prediction tools on the different data sets. "Unseen Data" includes both the Test data set and the (unseen parts of) Data Set 1-4.

Data Set (size)	TAW	EurOtop	NN	NNb	XGB	XGBr
Training data set (6943)	1.089	1.313	0.490	0.566	0.098	0.097
Test data set (1736)	0.995	1.207	0.478	0.580	0.284	0.285
Overtopping db (8679)	1.071	1.292	0.488	0.569	0.154	0.154
Data Set 1a (206)	0.631	0.696	0.958	0.394	0.349	0.403
Data Set 1b (156)	1.069	1.203	0.860	0.594	0.408	0.419
Data Set 2 (51)	1.047	1.266	0.714	0.575	0.448	0.356
Data Set 3 (242)	1.033	1.097	1.112	0.921	1.127	0.622*
Data Set 4 (177)	1.791	1.792	1.472	1.732	1.743	0.972*
Unseen data	1.170	1.232	1.104	1.005	0.723	0.411
All data	1.086	1.283	0.602	0.636	0.409	0.248

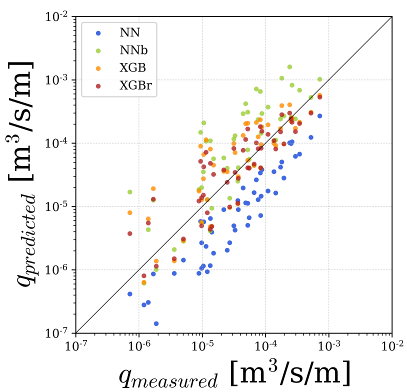
*Data points used in model training have been excluded from RMSE determination.



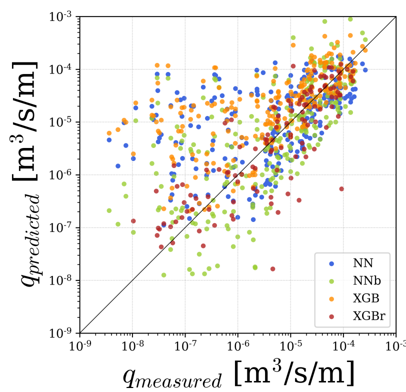
(a) DS1a Constant slope roughness.



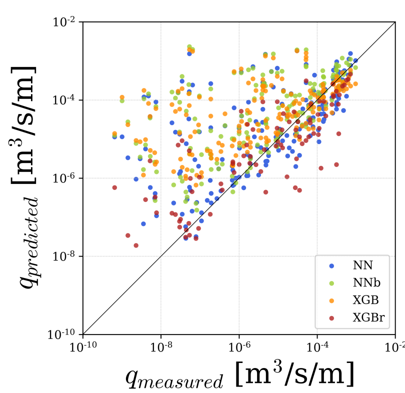
(b) DS1b Composite slope roughness.



(c) DS2 Crest walls.



(d) DS3 Oblique waves and crest walls.



(e) DS4 Oblique waves and berms.

Figure 5: Predictions by the NN (blue), NNb (green), XGB (orange) and XGBr (red) models for all additional test data sets.

Preprint, accepted for publication in Coastal Engineering: `https://doi.org/10.1016/j.coastaleng.2020.103830`

In Figure 5, the predictions by different machine learning models for the additional test data sets mentioned in Section 2.3 are shown. The RMSEs for all test data sets are listed in Table 5. The table lists the errors of both the different machine learning models and the TAW (2002) and EurOtop (2018) empirical formulae. Note that Data Sets 1-4 use $RF = 1$ and $CF = 1$ for the purpose of weighting in the RMSE calculation. Additionally, the RMSEs are based on all data with $q > 0$ [m^3/s/m], including very small discharges.

The predictions of the different machine learning models are shown in Figure 5a for Data Set 1a and Figure 5b for Data Set 1b. For the constant roughness cases in Data Set 1a, the predictions of the NN (blue) are in general significantly underestimating the overtopping discharge. In fact, the errors of the NN show a systematic behaviour in the sense that the predictions are rather constantly a factor of about 100 smaller than the measurements. Both the NNb (green) and XGB (orange) models also have a slight tendency towards underestimation, but less severe. This is reflected by the RMSE values in Table 5. The NNb (0.394) and XGB (0.349) models are fairly accurate, where the NN shows a much larger RMSE (0.958) for Data Set 1a due to the systematic differences.

Data Set 1b, featuring slopes with roughness differences, shows distinct clusters of points grouped in lines for both NN and NNb models. These distinct lines are formed by the different revetment types. This pattern suggests that the influence of roughness and roughness differences on the measured trends is not completely captured by the models. The XGB results exhibit a more random scatter along the diagonal, with a slight tendency towards overestimation. These observations are supported by the RMSE values, which compared to those for Data Set 1a, show some improvement of the NN model (0.860), a significantly worse performance for the NNb model (0.594), and a comparable performance for the XGB model (0.408). Presumably, the ability to recognize roughness differences through the f_{γ_f} parameter explains the relatively high accuracy of the XGB model for Data Set 1b.

Figure 5c shows the predictions for Data Set 2. The NN predictions are systematically underestimating the q. Conversely, the NNb and - to a lesser extent - the XGB model tend to slightly overestimate overtopping. This is mirrored in the RMSE values, where the XGB model is the most accurate (see Table 5).

All three models show a large amount of scatter for Data Set 3 (see Figure 5d), with errors of up to three orders of magnitude towards overestimating q. Notably, errors are significantly larger for small measured overtopping discharges ($q < 10^{-6}$m^3/s/m). Further analysis shows that the errors also increase with increasing angle of wave attack, β. The lower accuracy shown by all models (see Table 5) is likely the result of the small amount of training data containing $\beta > 0°$.

In Figure 5e, a pattern similar to Data Set 3 emerges for Data Set 4. All predictive models give very poor predictions, with a tendency towards overestimation of q up to four orders of magnitude. Again, the RMSE increases with both increasing β and decreasing q.

4.3. Retrained XGB model

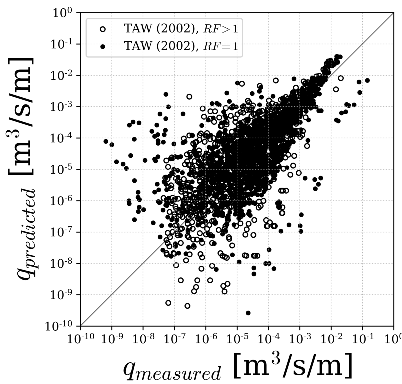
The combination of the lower accuracy on the additional test data sets containing oblique wave attack - Data Set 3 and 4 - and the fact that entries with $\beta > 0°$ are underrepresented in the training data suggests that expanding the training data with oblique wave data could improve the performance of data-driven models. To that end, the training data set for the XGB model is expanded by adding a random selection of half of Data Set 3 and 4. Subsequently, the model is retrained, again following the bootstrap resampling approach detailed in Section 3.3. The predictions of the retrained XGB model, indicated by "XGBr", are shown in red in Figure 5 and the associated errors are again listed in Table 5. Note that the predictions and RMSE shown are only based on the parts of Data Set 3 and 4 not used for model training.

By changing the training data set and retraining a data-driven model, its predictions change. The XGBr model shows significantly decreased errors for (the unseen parts of) Data Set 3 and 4, as expected. Additionally, the RMSE for Data Set 2 decreases as well, potentially because a part of the data added to the training data set also includes crest walls. The errors for Data Sets 1a and 1b slightly increase however. The reason will be that adding data to the training data set causes data similar to Data Set 1a and 1b to become relatively less important in model training. In general, the fact that the XGB models perform well on unseen data (both the test data set and Data Set 1-4) strongly suggests that the models are not overfitted and can be generically applied.

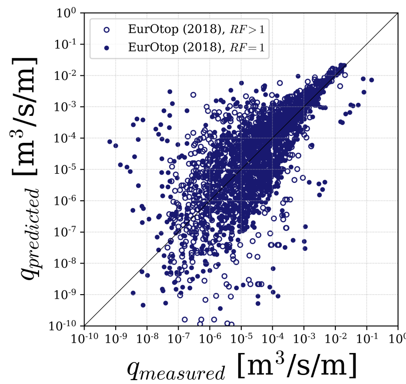
Comparison between the different machine learning methods shows that the XGB errors are generally smaller than NN and NNb for both the test and training data sets and the parts of the unseen data that are well represented in the training data (Data Set 1a, 1b and 2). Unseen data in data sparse regions of the training data set (Data Set 3 and 4 containing oblique waves) results in significantly larger errors. Expanding the training data set with oblique wave data and retraining the model (XGBr) results in significantly smaller errors for Data Set 3 and 4, at the cost of a small increase of the errors for Data Set 1a and 1b.

The results of the TAW (2002) and EurOtop (2018) empirical overtopping formulae are also included in Table 5. Note that here all data has been used, also data points that are not strictly within the validity range of the empirical expressions. This is done both to compare their performance on the same data set as the machine learning methods and because these empirical expressions are often applied outside their range of validity. TAW (2002) results in an RMSE of around 1 for most tests, with a higher accuracy for Data Set 1a and a lower accuracy for Data Set 4. A similar trend emerges from the EurOtop (2018) results, although the errors are slightly higher than for TAW. In general, the machine learning methods (both NN and XGB variants) perform better than the empirical overtopping formulae.

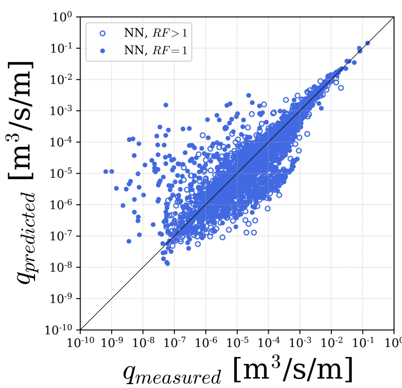
Finally, in the scatter plots of Figure 6 predictions by the different prediction methods are shown for the combination of the test data set and the additional test data sets (i.e. all data that has not been used to train the machine learning methods). Both the TAW (Figure 6a) and EurOtop (Figure 6b) empirical formulae show a large amount of scatter, with many outliers severely underestimating the amount of overtopping. Note that these outliers are a mix of both very reliable ($RF = 1$) and reasonably reliable ($RF = 2$ or 3) data. The NN (Figure 6c) and NNb (Figure 6d) models show less scatter, more or less symmetrically distributed around the diagonal. The XGB (Figure 6e) and XGBr (Figure 6f) models exhibit a very limited amount of scatter. These observations are reflected in the RMSE values in Table 5. In general, the XGB methods result in the smallest RMSE on all data. They are followed by both the NN and NNb models, that perform reasonably similar to each other. Lastly, the empirical formulae result in the largest errors, with TAW (2002) having a higher accuracy than EurOtop (2018).



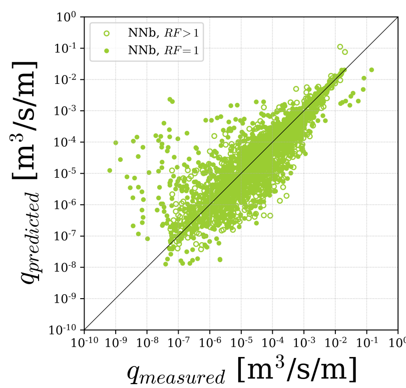
(a) TAW (2002).



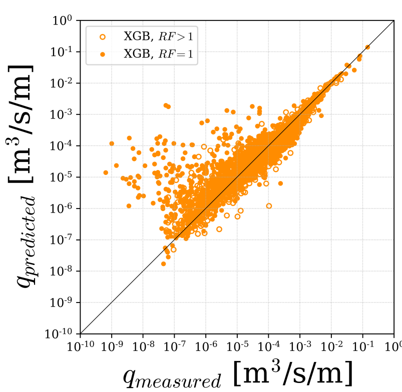
(b) EurOtop (2018).



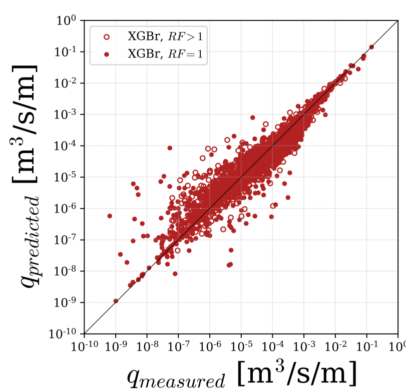
(c) NN.



(d) NNb.



(e) XGB.



(f) XGBr.

Figure 6: Overview of predictions for the test data set and additional test data sets using different prediction methods. For XGBr, only Data Set 3 and 4 data points not used in model training are shown.

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

5. Discussion

Section 4.3 shows that expanding the training data set with new data can greatly improve the overall performance of data-driven methods. This is especially true when newly added data covers parameter combinations that are currently not covered by, or underrepresented in the training data. Here this is the case for oblique wave attack combined with either a berm or a crest wall. Continuous expansion of training data and retraining and revalidation of models is recommended for data-driven methods. Another advantage of adding data from recent physical models is the relatively high reliability of recent data, e.g. due to more advanced reflection compensation techniques and second-order wave generation that are often lacking in older data.

In Section 2.2, a strict split was imposed between training and test data sets to convincingly demonstrate the predictive quality of the trained model. The NN by Van Gent et al. (2007), however, does not use a separate test data set. Instead, all data is used in the model training process. Due to the application of bootstrap resampling (as described in Section 3.3) the overall model is based on 500 individual models where no single model is trained on the entire training data set. For the sake of completeness, an XGB model is trained using the same method (see Figure 7). As a consequence of its construction, this model shows small RMSEs for both the overtopping database (0.100) and all data (0.092).

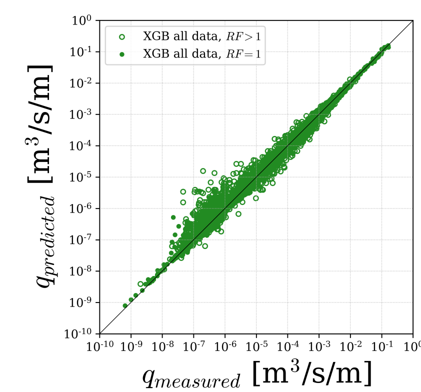


Figure 7: Overview of predictions for both the overtopping database and the additional test data sets by the XGB model trained on a bootstrap resampling of all data, following the same method as used in Van Gent et al. (2007).

In the model validation effort presented in this work, multiple data-driven overtopping prediction methods - including the new XGB model - are com-

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

pared to empirical overtopping formulae. Overall, the data-driven methods (especially the XGB models) perform better than the empirical formulae on both the overtopping database and the additional test data sets (unseen data) examined in this paper. This suggests that data-driven methods should become increasingly important as a tool for engineering and design of coastal structures, at least alongside, if not instead of, the existing empirical formulae. If, for design purposes, some conservatism is desirable, this could be derived from the confidence intervals that are given together with the predictions.

6. Conclusions and recommendations

In this work, the application of XGBoost to the prediction of mean wave overtopping is further refined and compared to other available prediction methods. The selection of features on which to train the model is expanded upon in detail, with significant improvements compared to existing literature. A combination of bootstrap resampling of the overtopping database and suitable selection of model hyperparameters results in realistic confidence intervals. All considered prediction methods are extensively validated on the training and test data sets. The XGBoost model outperforms other prediction methods on both test and training data sets from the overtopping database. All data-driven methods show less accuracy on the oblique wave data present in the additional test data sets, presumably because these cases are underrepresented in the overtopping database. Adding a randomly selected part of the new oblique wave data to the training data greatly improves the quality of the XGBoost model.

Similar to the lack of oblique wave data, the overtopping database contains many more white spots. For further research, it is recommended to identify these white spots and add data that falls within them. Hence, the white spots in the overtopping database can also be used to identify which data is useful to generate in new physical model experiments. At the same time, or as an alternative to physical model data, it is recommended to explore the possibility of adding numerical model data to the training data set. Numerical models prove to be a relatively efficient way of generating large amounts of data to address white spots in the training data set. Note however that this requires numerical models that are extensively validated and calibrated on physical model data, in order to obtain reliable numerical data.

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

References

Besley, P., Reeves, M., & Allsop, N. W. H. (1993). *Random wave physical model tests: overtopping and reflection performance*. Technical Report HR Wallingford. Report IT 384.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *CoRR*, *abs/1603.02754*. URL: http://arxiv.org/abs/1603.02754. arXiv:1603.02754.

Chen, W., Marconi, A., Van Gent, M. R. A., Warmink, J. J., & Hulscher, S. J. M. H. (2020a). Experimental study on the influence of berms and roughness on wave overtopping at rock-armoured dikes. *Journal of Marine Science and Engineering*, *8*. doi:10.3390/jmse8060446.

Chen, W., Van Gent, M. R. A., Warmink, J. J., & Hulscher, S. J. M. H. (2020b). The influence of a berm and roughness on the wave overtopping at dikes. *Coastal Engineering*, *156*, 103613. doi:https://doi.org/10.1016/j.coastaleng.2019.103613.

Deltares (). Overtopping neural network webtool. URL: https://www.deltares.nl/en/software/overtopping-neural-network/ (accessed on 8 April 2020).

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*. doi:https://doi.org/10.3390/w12061703.

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. London: Chapman & Hall.

ELI5 (). ELI5 python package. URL: https://github.com/TeamHG-Memex/eli5.

EurOtop (2018). *Manual on wave overtopping of sea defences and related structures*. J. W. van der Meer, N. W. H. Allsop, T. Bruce, J. de Rouck, A. Kortenhaus, T. Pullen, H. Schüttrumpf, P. Troch, B. Zanuttigh (Eds.), www.overtopping-manual.com.

Fisher, A., Rudin, C., & Dominici, F. (2018). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. arXiv:http://arxiv.org/abs/1801.01489.

Jacobsen, N. G., Van Gent, M. R. A., Capel, A., & Borsboom, M. (2018). Numerical prediction of integrated wave loads on crest walls on top of rubble mound structures. *Coastal Engineering*, *142*, 110 – 124. URL: http://www.sciencedirect.com/science/article/pii/S037838391730220X. doi:https://doi.org/10.1016/j.coastaleng.2018.10.004.

Lim, S., & Chi, S. (2019). Xgboost application on bridge management systems for proactive damage estimation. *Advanced Engineering Informatics*, *41*, 100922. doi:https://doi.org/10.1016/j.aei.2019.100922.

Lykke Andersen, T., Skals, K. T., & Burcharth, H. F. (2008). Comparison of homogenous and multi-layered berm breakwaters with respect to overtopping and front slope stability. In *Proc. 31th ICCE*. ASCE. doi:https://doi.org/10.1142/9789814277426_0273.

Oumeraci, H., Kortenhaus, A., & Burg, S. (2007). *Investigations of wave loading andovertopping of an innovative mobile flood defence system: Analysis of model testsand design formulae.* Technical Report Technische Universität Braunschweig,Leichtweiß-Institut für Wasserbau, Abteilung Hydromechanik und Küsteningenieurwesen. Report Nr. 949.

Steendam, G. J., Van der Meer, J. W., Verhaeghe, H., Besley, P., Franco, L., & Van Gent, M. R. A. (2004). The international database on wave overtopping. In *Proc. 29th ICCE, Vol.4* (pp. 4301–4313). World Scientific. doi:https://doi.org/10.1142/9789812701916_0347.

TAW (2002). *Wave run-up and wave overtopping at dikes.* Technical Report Technical Advisory Committee for Flood Defence in the Netherlands (TAW). Delft.

Van Doorslaer, K., De Rouck, J., Audenaert, S., & Duquet, V. (2015). Crest modifications to reduce wave overtopping of non-breaking waves over a smooth dike slope. *Coastal Engineering*, *101*, 69–88. doi:https://doi.org/10.1016/j.coastaleng.2015.02.004.

Preprint, accepted for publication in Coastal Engineering: https://doi.org/10.1016/j.coastaleng.2020.103830

Van Gent, M. R. A. (2020). Influence of oblique wave attack on wave overtopping at smooth and rough dikes with a berm. *Coastal Engineering*, *160*, 103734. doi:https://doi.org/10.1016/j.coastaleng.2020.103734.

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*, 586–593. doi:https://doi.org/10.1016/j.coastaleng.2006.12.001.

Van Gent, M. R. A., & Van der Werf, I. M. (2019). Influence of oblique wave attack on wave overtopping and forces on rubble mound breakwater crest walls. *Coastal Engineering*, *151*, 78–96. doi:https://doi.org/10.1016/j.coastaleng.2019.04.001.

Van Rossum, G. (1995). *Python tutorial*. Technical Report CS-R9526 Centrum voor Wiskunde en Informatica (CWI).

Victor, L., & Troch, P. (2012). Wave overtopping at smooth impermeable steep slopes with low crest freeboards. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, *138*, 372–385. doi:https://doi.org/10.1061/(ASCE)WW.1943-5460.0000141.

Zanuttigh, B., Formentin, S. M., & Van der Meer, J. W. (2016). Prediction of extreme and tolerable wave overtopping discharges through an advanced neural network. *Ocean Engineering*, *127*, 7–22. doi:https://doi.org/10.1016/j.oceaneng.2016.09.032.

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., & Si, Y. (2018). A data-driven design for fault detection of wind turbines using random forests and xgboost. *IEEE Access*, *6*, 21020–21031. doi:https://doi.org/10.1109/ACCESS.2018.2818678.