

Article

Predictions of Wave Overtopping Using Deep Learning Neural Networks

Yu-Ting Tsai ^{1,*}  and Ching-Piao Tsai ^{2,*} 

¹ Bachelor's Program of Precision System Design, Feng Chia University, Taichung 407, Taiwan

² Department of Civil Engineering, National Chung Hsing University, Taichung 402, Taiwan

* Correspondence: yuttsai@fcu.edu.tw (Y.-T.T.); cptsai@nchu.edu.tw (C.-P.T.)

Abstract: Deep learning techniques have revolutionized the field of artificial intelligence by enabling accurate predictions of complex natural scenarios. This paper proposes a novel convolutional neural network (CNN) model that involves deep learning technologies, such as the bottleneck residual block, layer normalization, and dropout layer, to predict wave overtopping at coastal structures under a wide range of conditions. To optimize the performance of the CNN model, the hyperparameter tuning process via Bayesian optimization is used. The results of validation demonstrate that the proposed CNN model is highly accurate in estimating wave overtopping discharge from hydraulic and structural parameters. The testing accuracy of the overtopping predictions using a prototype dataset shows that the proposed CNN model outperforms those existing machine learning models. An example application of the CNN model is presented for predicting prototype overtopping considering various crest freeboards of coastal structures.

Keywords: wave overtopping; deep learning technique; convolutional neural network; bottleneck residual block; convolution layer; layer normalization; dropout layer; hyperparameter optimal tuning process; Bayesian optimization; bootstrap resampling



Citation: Tsai, Y.-T.; Tsai, C.-P.

Predictions of Wave Overtopping Using Deep Learning Neural Networks. *J. Mar. Sci. Eng.* **2023**, *11*, 1925. <https://doi.org/10.3390/jmse11101925>

Academic Editors: Maria Teresa Reis, João Alfredo Santos and Tiago Fazeres Ferradosa

Received: 7 September 2023

Revised: 28 September 2023

Accepted: 1 October 2023

Published: 5 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The accurate assessment of wave overtopping discharges is crucial for the appropriate design of coastal structures that can prevent severe coastal flooding. Wave overtopping at coastal structures has been studied extensively through various methods, including field investigations, scale model measurements, and numerical simulations [1–7]. The CLASH (Crest Level Assessment of Coastal Structures) project [8] conducted numerous physical modeling and prototype investigations of wave overtopping discharge at various types of coastal structures. Various empirical formulas are included in the EurOtop manual [9] for the estimation of wave overtopping discharge. On the other hand, robust artificial neural network (ANN) models, utilizing hydraulic and structural parameters sourced from either the new EurOtop database [9] or the original CLASH database [10] as input variables, have demonstrated efficient performance in predicting wave overtopping discharge at coastal structures [4,7,11–14]. Each test in the EurOtop database is characterized by a reliability factor (RF) and complexity factor (CF) [10], which are used to select reliable data for training and improving ANN models.

Previous studies have proposed a variety of multilayer architectures for ANN models in the field of machine learning. Verhaeghe et al. [7] proposed a two-phase neural network model, including a classifier–quantifier scheme, to distinguish negligible from significant overtopping and avoid large overpredictions in areas of low overtopping. However, Zanuttigh et al. [11] argued that classifier–quantifier schemes do not truly improve ANN performance and instead increase the complexity of the ANN architecture while creating undesirable discontinuities in predictions. den Bieman et al. [12] presented a new model using an advanced machine learning technique called XGBoost (XGB) associated

with feature engineering, which generally outperforms conventional ANN models and empirical formulas in terms of prediction accuracy. Bootstrap resampling techniques were employed in these machine learning models to evaluate the uncertainties associated with their predictions.

Accurately predicting wave overtopping discharge in complex scenarios remains a challenging task. As an alternative to machine learning artificial neural networks (ANNs), deep convolutional neural networks (CNNs) present a promising avenue to tackle this issue. Originally developed for identifying the characteristics of individual images from large datasets, CNNs have found applications in big data analysis and decision making [13,14]. Utilizing deep convolutional neural networks for the prediction of wave overtopping discharge represents a relatively recent research area, yet it has already exhibited significant potential. Through convolutional processing for feature extraction, CNNs can effectively capture the distinctive attributes of data representations and employ filters to accentuate the respective importance of each output task. As a result, CNNs may offer significant improvements in the prediction accuracy of wave overtopping discharge.

This article presents a CNN model using a serial bottleneck residual block architecture that incorporates convolution layers, layer normalization, and a dropout layer. The model is designed to determine the relevance of hydraulic and coastal structure inputs for accurately predicting wave overtopping discharge. This paper is organized as follows. Section 2 describes the new EurOtop database which is applied to train the CNN model. Then, we explain the model expression, network layer architecture, and hyperparameter optimization process used for training the CNN model. As applied in previous ANNs, the bootstrap resampling technique is also introduced in training the CNN model. Next, we discuss the accuracy of the CNN model predictions and compare them with those of existing ANN models. An example application of the model for predicting prototype overtopping is also presented. Finally, the conclusions of the study are addressed.

2. EurOtop Database

The proposed CNN model was trained using data from the new EurOtop database [9], which is an extensive dataset containing over 17,000 tests. This dataset includes nearly 13,500 records specifically related to wave overtopping discharge. This new EurOtop database is an extension of the original CLASH database, which was compiled from approximately 10,000 schematized tests conducted worldwide. By utilizing this comprehensive dataset, the CNN model can learn from a wide range of test scenarios and improve its predictive capabilities for wave overtopping discharge. Figure 1 depicts the general geometric and relevant hydraulic parameters of coastal structures. Full descriptions of the parameters, wave conditions, and structural cross sections in the database are provided in the literature [9,11,12,15,16]. The overtopping experiments in the database are categorized into seven groups based on the structure type and oblique wave attack conditions [9]; they are rock permeable straight slopes (group A), rock impermeable straight slopes (group B), armor units with straight slopes (group C), smooth and straight slopes (group D), structures with combined slopes and berms (group E), vertical walls (group F), and oblique wave attacks (group G).

The parameters selected for training the CNN model are listed in Table 1 and comprise three hydraulic parameters of waves and thirteen structural parameters as inputs and one parameter (wave overtopping discharge) as the output. The training dataset of the neural network model consists of data derived from laboratory tests conducted at various scales. To ensure the applicability of the model to prototype scenarios, the use of dimensionless parameters is advantageous. Following [9,11,16], the significant wave height and water depth at the structure toe are represented as the wave steepness $H_{m0,t}/L_{m-1,0,t}$ and the relative water depth $h/L_{m-1,0,t}$, respectively, by dividing the wavelength $L_{m-1,0,t}$ in which $L_{m-1,0,t}$ is calculated by $1.56T_{m-1,0,t}^2$, and $T_{m-1,0,t}$ is the spectral wave period at the structure toe. The parameters B_t , G_c , and B describing horizontal measures of structure widths are normalized by the wavelength $L_{m-1,0,t}$ and the parameters h_t , D_u , D_d , A_c , R_c , and

h_b describing vertical measures of structure heights are normalized by the wave height $H_{m0,t}$. By incorporating dimensionless parameters, the neural network model can effectively capture the underlying physics and behavior of wave overtopping, regardless of the specific scale of the experiments.

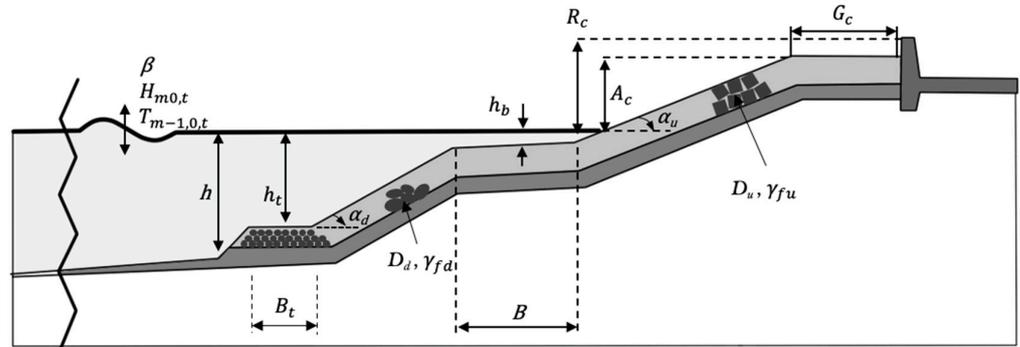


Figure 1. Schematic of structure based on CLASH, including geometric and hydraulic parameters [9]. $H_{m0,t}$: spectra significant wave height at the structure toe; $T_{m-1,0,t}$: spectral wave period at the structure toe; h : water depth at the structure toe; h_t : toe submergence; B_t : toe width; B : berm width; h_b : berm submergence; A_c : crest freeboard of armor structure; R_c : crest freeboard of wall; G_c : crest width of armor structure; α_u and α_d : angles of armor structure slope part above/below the berm; D_u and D_d : sizes of the armor elements along up/down slopes; γ_{fu} and γ_{fd} : roughness factors for up/down slopes; β : incident angle of wave direction.

Table 1. CNN input and output parameters.

#	Parameters	Definition of the Parameters	
1	Hydraulic parameters of waves	$H_{m0,t}/L_{m-1,0,t}$	Wave steepness at the structure toe
2		$h/L_{m-1,0,t}$	Relative water depth at the structure toe
3		β	Wave obliquity
4	Structure parameters	$h_t/H_{m0,t}$	Relative submergence of the toe structure
5		$B_t/L_{m-1,0,t}$	Relative width of the toe structure
6		$\cot\alpha_d$	Cotangent of the structure slope below the berm
7		$\cot\alpha_u$	Cotangent of the structure slope above the berm
8		γ_{fd}	Roughness factor for $\cot\alpha_d$
9		γ_{fu}	Roughness factor for $\cot\alpha_u$
10		$D_d/H_{m0,t}$	Relative size of the armor elements along $\cot\alpha_d$
11		$D_u/H_{m0,t}$	Relative size of the armor elements along $\cot\alpha_u$
12		$A_c/H_{m0,t}$	Relative crest freeboard of armor structure
13		$R_c/H_{m0,t}$	Relative crest freeboard of wall
14		$G_c/L_{m-1,0,t}$	Relative crest width of armor structure
15		$B/L_{m-1,0,t}$	Relative berm width
16		$h_b/L_{m-1,0,t}$	Relative berm submergence
1	Predicted parameter	q_s	Normalized wave overtopping discharge per unit width

Following [11,16], the wave overtopping discharge q in the database is scaled by a normalized formula for training, which is defined as:

$$q_s = \frac{\log_{10}(q_{ad}) - \min[\log_{10}(q_{ad})]}{\max[\log_{10}(q_{ad})] - \min[\log_{10}(q_{ad})]} \quad (1)$$

in which q_{ad} is defined as

$$q_{ad} = \frac{q}{\sqrt{gH_{m0,t}^3}} \quad (2)$$

where q is the original mean wave overtopping discharge value from the EurOtop database and g is the gravitational acceleration (9.8 m/s^2). Equation (2) represents a dimensionless formula known as the relative wave overtopping discharge. As elucidated in [13], the application of Equation (1) for normalization confines the fitted data within a range of 0 to 1, thereby mitigating the scale-dependent nature of the error. Subsequently, the predicted output value q_s generated by the trained CNN is reverted to obtain the corresponding value of q .

A weight factor $WF = (4 - RF) \cdot (4 - CF)$ defined in [4] was adopted in previous works for the selection of reliable data to minimize the influence of environmental uncertainties and irregular structures. Higher WFs for the overtopping database indicate greater reliability and simpler structural schemes. As the same in earlier prediction methods [11,16], excluding the data of very unreliable ($RF = 4$) or complex ($CF = 4$), $q = 0$, and missing values, a total of 8653 records labeled as core data in the overtopping database are used in the proposed CNN model. It is noted that the core data excluded the dataset of prototype overtopping.

3. Method Description

3.1. Model Description

In this section, we present how the proposed CNN model learns the inputs of the hydraulic and structural parameters corresponding to the output of wave overtopping discharge. The overtopping database contains many different structure types and related input parameters, but they may have the same distributions in the output parameters, as known as a covariate shift problem. The general neural network model may require specific feature engineering to select appropriate input parameters, preventing the model from possibly misclassifying the data. Alternatively, the CNN gains experience by observing a training set of input data and predicting the output without any feature assumptions. Accounting for this problem, the convolutional layer [13], layer normalization [17], and a dropout layer [18] are used for the proposed CNN, described as follows.

Convolutional Layer: This layer operates by traversing a compact filter across the input data, enabling the detection of patterns and distinctive features. The equation of the convolutional layers is expressed as follows:

$$\mathbf{x}^{(l)} = \mathbf{w} \cdot \mathbf{x}^{(l-1)} + \mathbf{b} = \sum_{\tau=0}^T [w[\tau]^{(l)} x[t - \tau + \phi]^{(l-1)}] + b[t] \quad (3)$$

where the superscript l represents the index of each neural network layer and t denotes the index of the input parameters. In Equation (3), $x^{(l-1)}$ is a given input dataset, w is a set of one-dimensional convolutional filter coefficients, and ϕ denotes the shift number. Equation (3) can be described as the area under the function $w[\tau]$, which is weighted by the function $x[-\tau]$, which is shifted by an amount t . As t changes, the weighting function $x[t - \tau]$ emphasizes different components of the input weighting function $w[\tau]$ to extract the significant features determined by the filter matrices. $b[t]$ is the bias coefficient as the constant of a linear function.

Layer normalization layer: The main objective of layer normalization is to reduce the internal covariate shift that occurs during training, which helps stabilize the distribution of the inputs to a layer. Layer normalization works by normalizing the values within a layer, making their mean close to zero and their variance close to one. This normalization can speed up training and allow for the use of larger learning rates. Additionally, it helps with better gradient flow during backpropagation, which can lead to faster convergence and more stable training.

Dropout layer: It is used to prevent overfitting in neural networks. During training, a dropout layer randomly disables a fraction of the neurons or units in the layer by setting them to zero for that particular pass. By introducing noise and preventing neurons from relying too heavily on any particular input, dropout promotes learning more robust and

generalized features. This regularization technique helps prevent overfitting and improves the generalization of neural networks, making them more capable of performing well on unseen data.

The proposed CNN consists of an input layer for inputting sequence data such as the parameters shown in Table 1, hidden layers, and an output layer as the fully connected layer to calculate a weighted sum of its inputs and is used to make predictions q_s . In any feed-forward neural network, the middle layers are referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. To make a deeper CNN model, a bottleneck residual block [19] involving convolution layers, layer normalization, and a dropout layer are applied to mitigate degradation (accuracy saturation) and the overfitting problem [20] caused by combining many hidden layers. It may also accelerate learning for our proposed CNN. For the architecture of the overall connection model with the aforementioned network representations, the bottleneck residual block adds a route for simple addition into each hidden layer connection as shown in Figure 2.

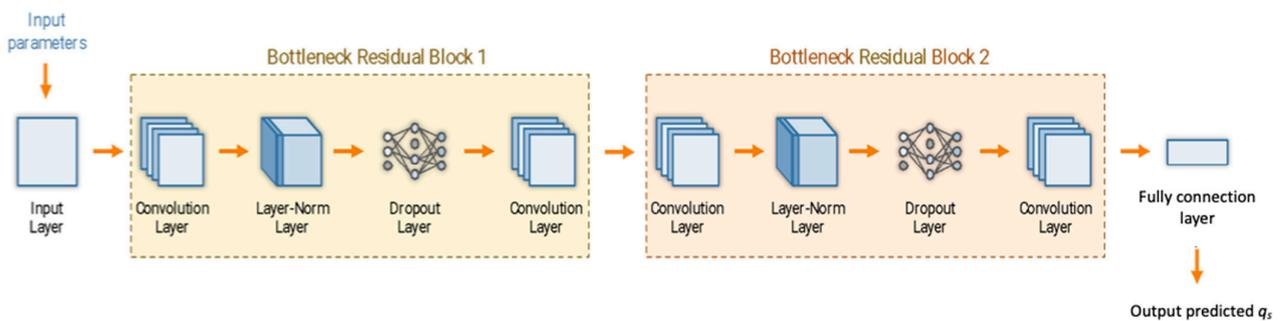


Figure 2. Architecture of proposed CNN with the combination of bottleneck residual blocks including convolutional, layer normalization, and dropout layers in serial for learning relevant feature values from inputs for output of wave overtopping discharge q_s .

3.2. CNN Training with Hyperparameter Optimization

To obtain an optimal model, it is common practice to limit the number of learning iterations or set a stopping threshold for the objective loss function in order to prevent overfitting. In this study, we employ the hyperparameter optimal tuning process (HOTP) [21,22] to minimize the objective loss function and mitigate overfitting [20]. The objective loss function combines the mean square loss function [23] and the Wasserstein distances W [24] between two measures $P_{\hat{q}}$ and P_q , which are the cumulative distribution functions of the model predictions \hat{q} and the observed values q , respectively. The objective loss function L can be mathematically defined as follows:

$$L = \lambda_1 \frac{1}{I} \sum_{i=1}^I \sqrt{(\hat{q}_i - q_i)^2} + \lambda_2 W_s(P_{\hat{q}}, P_q) \tag{4}$$

In Equation (4), I is the subsample number to be randomly extracted from the training set. Each summand L is associated with the n -th observation in the subsample during training. Here, λ_1 and λ_2 are constraints on the sensitivity of the squared error loss and Wasserstein distances $W_s(P_{\hat{q}}, P_q)$. To train the CNN, the adaptive movement estimation algorithm, a backward propagation-based iterative method developed by Adam [25], is employed for optimizing the objective function with desirable smoothness properties. The learning rate is one of the key hyperparameters when training a deep learning neural network model. It is determined through HOTP to find the value that results in the best model performance.

Figure 3 provides an overview of the hyperparameter optimization (HOTP) process utilized to develop an optimal CNN model. The primary objective of HOTP is to identify the hyperparameters that yield the best performance of the CNN model, while avoiding

overfitting. It is important to emphasize that the validation dataset used during HOTP is distinct from the testing dataset and is not part of the training process.

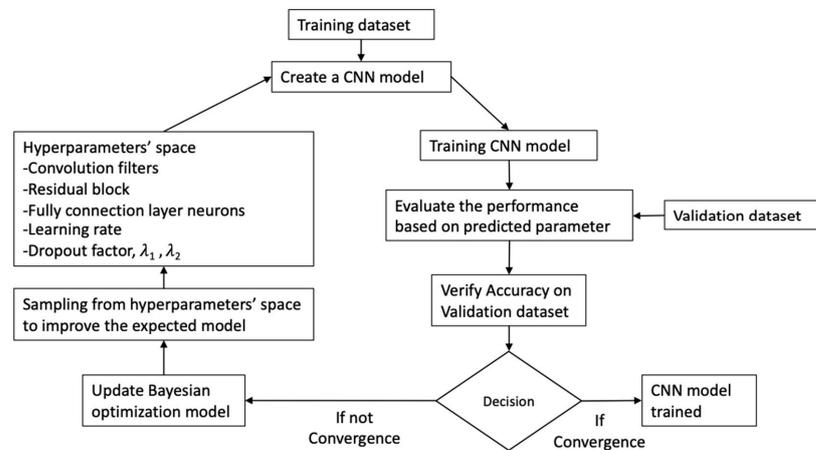


Figure 3. Schematic of hyperparameter optimal tuning process (HOTP) to create the CNN model.

In each iteration of HOTP, a new set of hyperparameters is evaluated by computing the associated loss using the validation dataset. The objective function aims to minimize this loss, enabling the identification of optimal hyperparameters that lead to a model with the lowest possible loss. Bayesian optimization [26] is employed in this study to conduct an exhaustive search of the hyperparameter space, focusing on a predetermined subset of hyperparameters. The optimization process involves exploring different combinations of hyperparameters based on previous observations, with the goal of refining the model’s performance.

Once the CNN model is trained using HOTP and the optimal hyperparameters are determined, it can be utilized to make predictions for overtopping values. This comprehensive approach ensures that the CNN model is fine-tuned to achieve optimal performance and can provide accurate predictions for the given task.

To configure hyperparameters for a CNN, first we define the network’s architectural aspects according to Figure 2 and initialize the corresponding hyperparameters randomly, such as the number of layers, filters, and size of neurons. Secondly, set general hyperparameters such as the neurons of convolution layers and fully connected layers, number of residual blocks, learning rate, batch size, dropout rate, and weighting coefficients of the loss function to fine-tune the training process. After establishing these foundational settings, fine-tune the hyperparameters with HOPT on a validation set. Table 2 displays a hypothetical interval of the hyperparameter space used during the HOTP process. The aim of HOTP is to identify the optimal values for these hyperparameters, which are crucial for the performance of the CNN model. Within Table 2, the optimal values of the hyperparameters are denoted by a star (*) symbol, indicating the configurations that yield the best performance for the CNN model.

Table 2. Setup of hypothetical interval of hyperparameters for HOTP.

Hyperparameter Name	Values	Definition of the Hyperparameter
Convolution_filters	[16, 24, 32, 64, 128, 256 *]	Number of filters w in each convolution layer
Residual_block	[1, 2 *]	Number of residual block
Fully_connection_layer_neurons	[32, 64, 128 *, 256, 512]	Neurons in a fully connected layer
Learning_rate	[0.0001 *, 0.0002, ..., 0.1]	Given range of learning rate for CNN training
Dropout_rate	[0.1, 0.2, ... 0.7 *, 0.8, 0.9],	Determines the probability of a neuron being dropped out
λ_1	[0.5, 0.6, ..., 1.0 *]	Given weights of RMSE term of L
λ_2	[0.1, 0.2, ..., 0.5 *, ..., 1.0]	Given weights for constraint term of L

4. Results and Discussion

4.1. Training Setup

For the CNN model in this study, the training process was carried out with specific configurations. In the context of neural network training, an epoch refers to a complete iteration over the entire training dataset. During an epoch, the model goes through multiple iterations, each involving a batch of training samples. The number of iterations per epoch is determined by the batch size. In this study, each batch size was limited to a maximum size of 256, while the remaining samples constituted the minimum quantity for the last batch size. The training was performed with a maximum of 5000 epochs, and the early stopping algorithm was employed to prevent overfitting. This algorithm continuously monitored the validation performance and stored the model with the best results when a predefined global training accuracy was achieved.

The training procedure took place on a computer equipped with a 3.4 GHz Intel Core i7-6700 CPU and two NVIDIA GeForce GTX 1080 Ti GPUs. The implementation of the CNN model was conducted using the Python programming language in conjunction with the TensorFlow library [27]. This setup ensured efficient training and allowed for the attainment of optimal outcomes.

In this study, the entire core dataset of the EurOtop overtopping database was utilized, following the approach outlined in [12]. The database was randomly divided into two portions: 80% for training the CNN model and 20% for validation. Similar to previous machine learning models [4,11,12,16], we employed the bootstrap resampling technique for the training of the CNN model. The bootstrap resampling technique involves creating N resamples from the original database. Each resample forms a distinct training and validation dataset. Subsequently, the CNN model is trained using each resample dataset, and the predicted overtopping discharge is obtained by calculating the ensemble mean of the predictions. This technique allows for the evaluation of the model's performance under various training conditions and provides a robust estimation of its predictive capabilities.

When dealing with a substantial volume of data, increasing the number of iterations in bootstrap resampling typically results in more precise estimates derived from these resamples [28]. However, in the context of the overtopping database, which contains relatively modest-sized samples compared to fields with datasets comprising hundreds of thousands of data points or more, excessively increasing the number of bootstrap resampling iterations may not yield substantial improvements in predicting future outcomes. Moreover, this practice can potentially create a deceptive sense of certainty if each model within the ensemble exhibits notable levels of uncertainty.

In cases where a model already displays minimal variance in its prediction outcomes, reducing uncertainty can be accomplished by examining only a limited number of bootstrap samples. This approach not only aids in reducing training time but has also proven to be effective. Therefore, it becomes crucial to determine the appropriate number of bootstrap resamples required to construct a CNN model ensemble. In this study, we evaluate the model's performance across various numbers of bootstrap resamples, ranging from $N = 1$ to $N = 500$. It is noteworthy that the scenario with $N = 500$ aligns with the same number of bootstrap resamples used in [4].

For the assessment of predicted quality, we employ the weighted root-mean-square error (RMSE) as the error criterion, as defined in [12]. The weighted RMSE is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{\sum_k^K WF_k} \sum_{k=1}^K WF_k \cdot [\log_{10}(\hat{q}_k) - \log_{10}(q_k)]^2} \tag{5}$$

where K is the observation number, and q_k and \hat{q}_k are the observed and predicted values, respectively. This error criterion takes into account the weights assigned to different variables or data points, allowing for a more comprehensive evaluation of the prediction

accuracy. By considering the weighted RMSE, we can assess the performance of the model in a more nuanced and representative manner.

In Figure 4, the variation of the weighted root-mean-square error (RMSE) is depicted from the ensemble mean with bootstrap resample numbers ranging from $N = 1$ to $N = 500$, along with the corresponding running time. It can be observed that as the bootstrap resample number increases, the RMSE decreases. The RMSE value converges at $N = 10$, as shown in Figure 4. Table 3 compares the RMSE values for $N = 1, 10$, and 500 , indicating that satisfactory accuracy can be achieved using $N = 10$. It is worth noting that the computational time is significantly different between $N = 10$ (44 seconds) and $N = 500$ (2155 seconds). This finding suggests that employing $N = 10$ bootstrap resamples balances accuracy and efficiency for the predictions. Figure 5 displays the predictions made by the trained CNN model using $N = 10$ for both the training and validation datasets. The proposed CNN model incorporated a strategy to mitigate overfitting, which was validated by the strong performance observed in Figure 5, where the training and validation results exhibited good agreement.

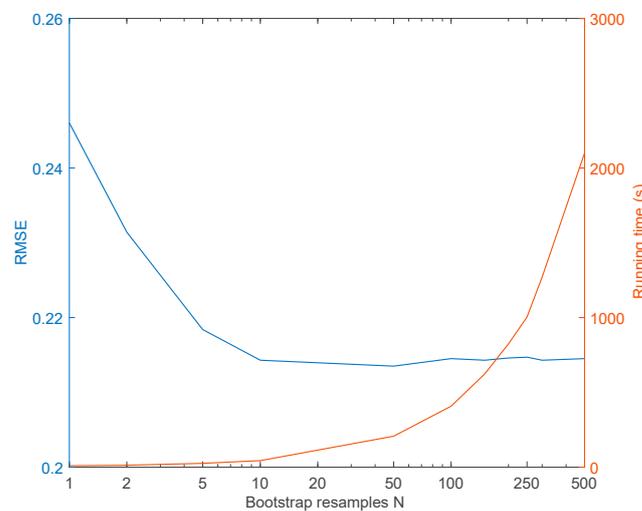


Figure 4. Comparisons of the RMSE and running time using different bootstrap resamples.

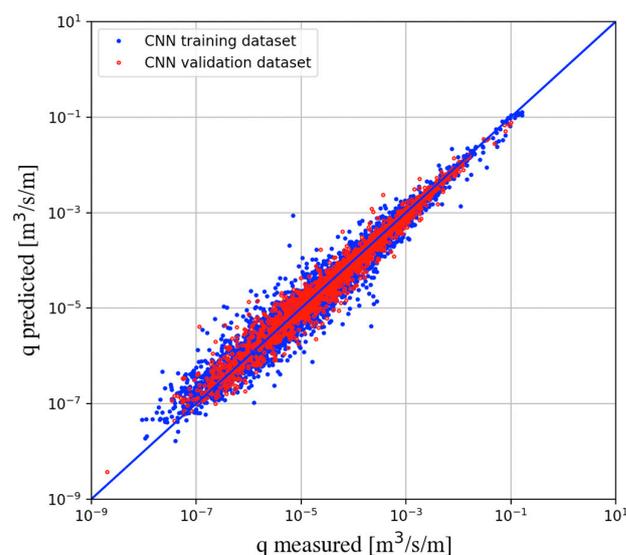


Figure 5. Predictions by the trained CNN for the training and the validation dataset with 10 bootstrap resamples.

Table 3. The comparison of predicted RMSEs for the training and validation datasets using different numbers of bootstrap resamples.

Dataset (Size)	Bootstrap Resamples		
	N = 1	N = 10	N = 500
Training data (6922)	0.238	0.230	0.225
Validation data (1731)	0.246	0.214	0.215

4.2. Verification on the Overtopping Database

Following the precedent of earlier ANN models [4,11,14], the trained CNN model is assessed for accuracy using the entire core dataset from the overtopping database. The trained CNN model can be accessed online at <https://github.com/fcuyttsai/overtoppingEve.git/> (accessed on 1 September 2023). The predictions of existing ANNs can be directly estimated from their respective websites. These existing ANN models are the NN model [12] (online at <https://www.deltares.nl/en/software/overtopping-neural-network/> (accessed on 1 May 2023)), the NNb model [11] (online at <http://overtopping.ing.unibo.it/overtopping/> (accessed on 1 May 2023)), and the XGB model [12] (online at <https://www.deltares.nl/en/software/overtopping-xgb/> (accessed on 1 May 2023)). In comparison to the deep learning CNN model, the NN, NNb, and XGB models fall within the realm of machine learning techniques. The NN model [4] employs the conventional back-propagation neural network as its learning method. Meanwhile, the NNb model [11], which utilizes an extended CLASH database as its training dataset, builds upon the fundamental NN model concept but integrates a quantitative classifier and three quantifiers. On the other hand, the XGB model [14], as an alternative to both the NN and NNb models, leverages an advanced machine learning technique known as XGBoost to predict overtopping discharge.

In Table 4, we present a comparison of the resulting weighted root-mean-square errors (RMSEs) for wave overtopping discharge predictions, along with the Pearson correlation coefficient (CC) and R² as measures of agreement. The Pearson correlation coefficient (CC) is given by:

$$CC = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y} \tag{6}$$

where x_i is the i -th value of observation, y_i is the i -th value of prediction, \bar{x} is the mean value of observation, \bar{y} is the mean value of prediction, s_x is the standard deviation of the observation, and s_y is the standard deviation of the prediction. R² is the square value of CC that is the coefficient of determination.

Table 4. The comparison of predicted RMSEs of different models.

Dataset (Size)	Model	RMSE	CC	R ²
Entire core data (8653)	NN	0.654	0.869	0.756
	NNb	0.639	0.858	0.736
	XGB	0.199	0.990	0.981
	CNN	0.112	0.996	0.991
A: rock permeable straight slopes (1131)	NN	0.702	0.768	0.590
	NNb	0.687	0.791	0.625
	XGB	0.247	0.977	0.955
	CNN	0.153	0.989	0.978
B: rock impermeable straight slopes (104)	NN	0.624	0.864	0.747
	NNb	0.823	0.710	0.504
	XGB	0.349	0.984	0.968
	CNN	0.136	0.994	0.989

Table 4. Cont.

Dataset (Size)	Model	RMSE	CC	R ²
C: armor units with straight slopes (934)	NN	0.506	0.872	0.760
	NNb	0.583	0.858	0.737
	XGB	0.219	0.983	0.966
	CNN	0.118	0.991	0.983
D: smooth and straight slopes (2069)	NN	0.480	0.972	0.946
	NNb	0.380	0.943	0.889
	XGB	0.120	0.996	0.991
	CNN	0.086	0.997	0.994
E: structures with combined slopes and berms (1631)	NN	0.784	0.838	0.703
	NNb	0.670	0.877	0.770
	XGB	0.204	0.987	0.975
	CNN	0.093	0.997	0.993
F: vertical walls (1677)	NN	0.685	0.863	0.745
	NNb	0.704	0.845	0.714
	XGB	0.206	0.988	0.977
	CNN	0.110	0.994	0.988
G: oblique wave attacks (1107)	NN	0.760	0.777	0.603
	NNb	0.872	0.772	0.595
	XGB	0.228	0.983	0.965
	CNN	0.128	0.989	0.978

As a result of the comparison, the RMSE value for the entire core dataset is 0.112 for the CNN model and 0.199 for the trained XGB. These values are significantly smaller than the RMSE values obtained from the trained NN (0.654) and NNb (0.639) models. Additionally, the CC and R² values further demonstrate that both the present deep learning model (CNN) and advanced machine learning model (XGB) outperform the conventional machine learning models (NN and NNb). The corresponding quantile–quantile plot of wave overtopping discharge predictions for each model is displayed in Figure 6.

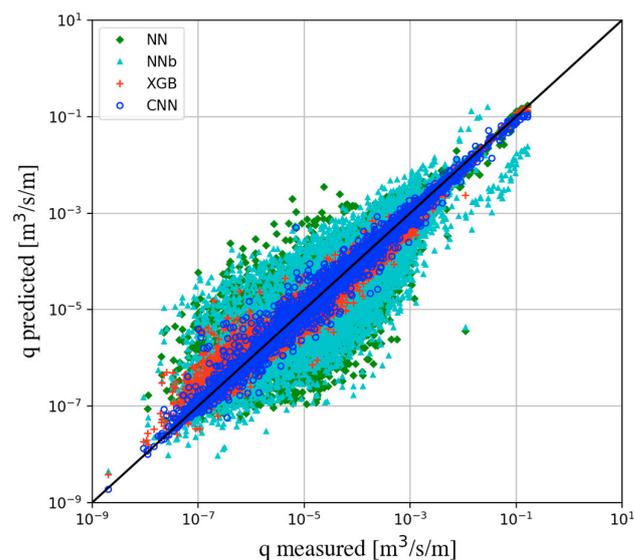


Figure 6. Comparison of predictions for the wave overtopping database predicted by the different models with the entire core dataset.

To demonstrate the adaptability of the current CNN model to different scenarios, we directly extract the results of Figure 6 for the groups A–G of the EurOtop database, based on structure type and oblique wave attack conditions. The predictions of each model are

plotted in Figure 7a–g. The scatterplots in Figure 7a–g show a high correlation between the predictions and observations for each group using the CNN model. The results indicate that all models perform relatively well for group D, which represents the situation of coastal structures with smooth and straight slopes. The CNN model also performs better for group B data, despite it containing only a few observations compared to other categories. Regarding the oblique wave attack prediction (group G), it is noted by [12] that the low accuracy of the NN models may be due to the limited amount of training data, leading to an improvement in the XGB model by incorporating new oblique wave cases into the training data. However, as shown in Table 4, the group G dataset comprises 1107 cases, which is larger than one-eighth of the total dataset. It can be observed that the current CNN model achieves highly accurate oblique wave attack (group G) predictions without the need for additional datasets.

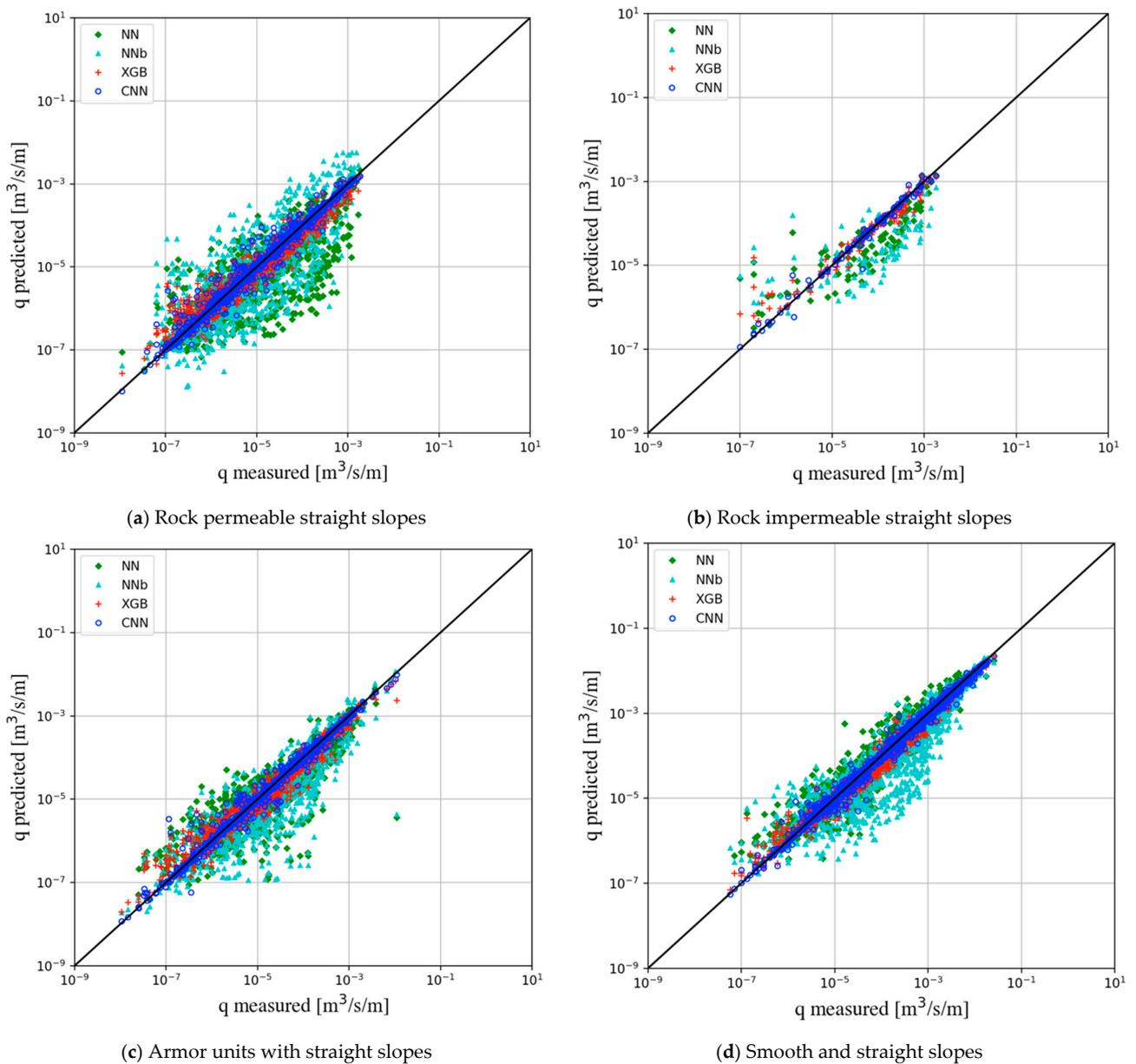


Figure 7. Cont.

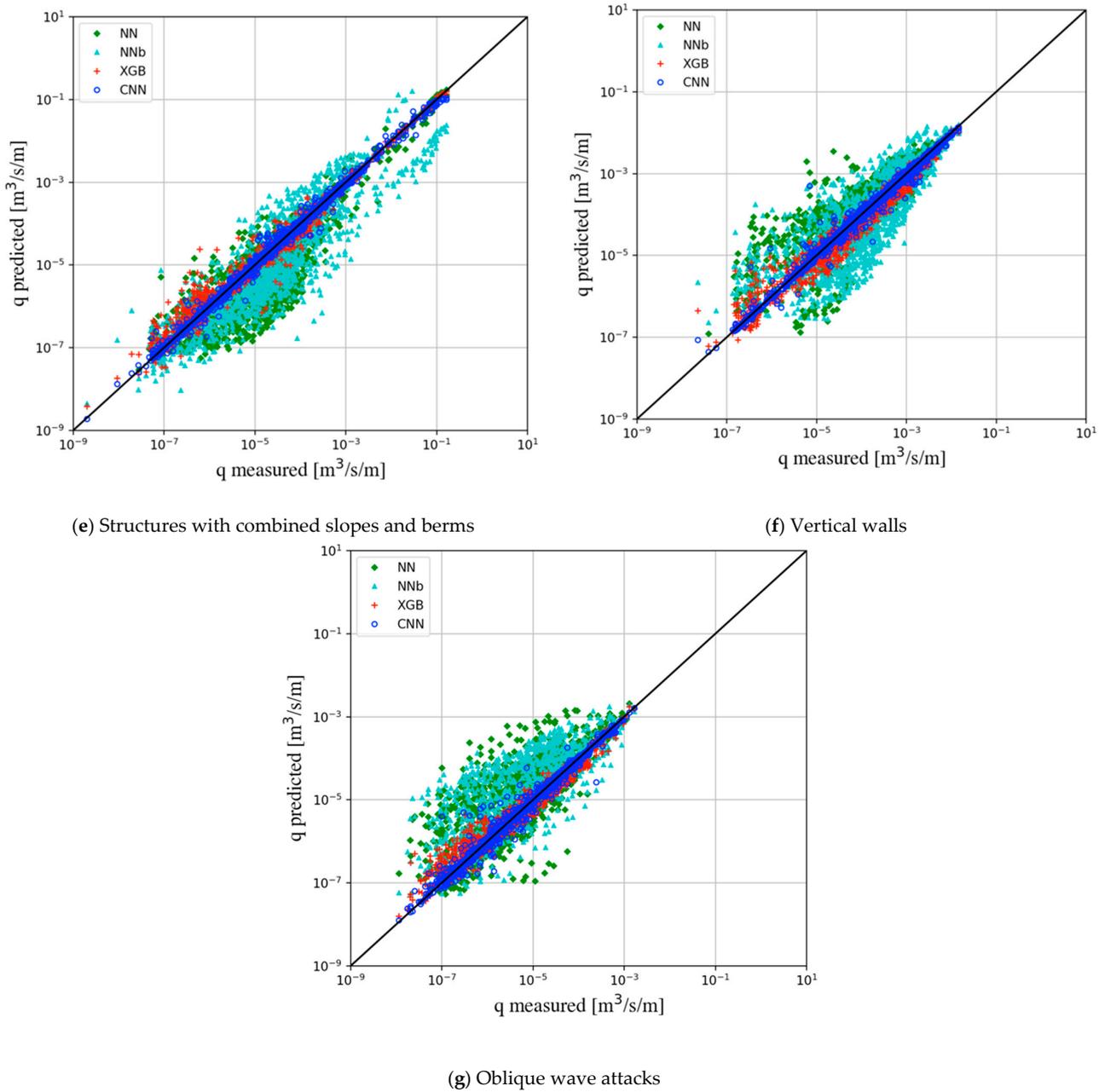


Figure 7. Comparisons of predicted wave overtopping discharges of different models for various coastal structures and oblique wave attack conditions with the entire core data.

4.3. Testing/Prediction Using Prototype Overtopping Dataset

Designers expect prediction models to be applied to prototype scenarios. The testing performance of the trained CNN model is evaluated using the prototype overtopping dataset, which was not seen during the model’s training and validation processes. The EurOtop database provides 150 wave overtopping measurement records from prototype situations. These records include 11 measurements labeled as C-59.1 to C-59.11 [28] from 9 storms between 1999 and 2004 at Zeebrugge breakwater, 77 measurements labeled as G-18.1 to G-18.77 [29] from 7 storms between 2003 and 2004 at Ostia breakwater, and 62 measurements labeled as F-18.1 to F-18.39 and G-12.1 to G-12.23 [30] from other sources.

The testing results of the trained CNN model are depicted in Figure 8, where the predictions of the NN, NNb, and XGB models are estimated directly from their respective websites using the same data inputs. It is important to note that all these models utilized the entire core dataset (i.e., laboratory measured data) during the training and validation

processes, with the prototype dataset remaining unseen. As illustrated in Figure 8 and Table 5, the CNN model exhibits better performance in testing when applied to the prototype dataset. It is worth noting that all models achieve high correlation coefficient (CC) values. However, due to the larger scale of the prototype, the RMSE values reported in Table 5 are understandably larger than those observed in the laboratory-scale data presented in Table 4. Furthermore, real-world prototype scenarios can introduce unanticipated variables such as noise, surface irregularities, wind, spray, currents, and the scale effect itself. These sources of uncertainty can contribute to more pronounced prediction errors.

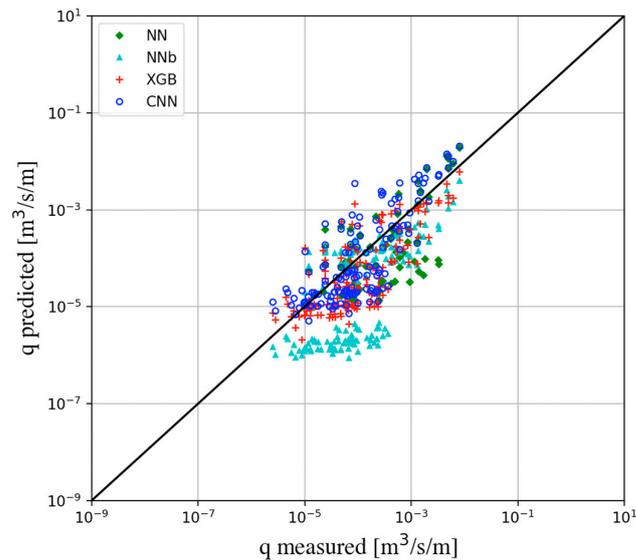


Figure 8. Comparison of predictions for the prototype wave overtopping discharge for different models.

Table 5. The comparison of predicted RMSEs of different models for the prototype dataset.

Dataset (Size)	Model	RMSE	CC	R ²
All testing data (150)	NN*	0.936	0.874	0.764
	NNb	1.218	0.895	0.802
	XGB	0.657	0.868	0.753
	CNN	0.555	0.932	0.868

* For the NN model, it is out of range for the estimation of G-18.1 to G-18.77. This result only calculates 23 cases of G-12.1 to G-12.23 for the NN model.

The prototype overtopping database can be divided into three categories: armor units with straight slopes (C-59.1 to C-59.11), vertical walls (F-18.1 to F-18.39), and oblique wave attacks (G-12.1 to G-12.23 and G-18.1 to G-18.77). Figure 9 presents the testing results for these three types of datasets, extracted directly from the results of Figure 9. As mentioned in [11], their model (i.e., NNb model) underestimated the predicted values of prototype overtopping for the Ostia breakwater (i.e., G-18.1 to G-18.77), while the predictions for Zeebrugge (i.e., C-59.1 to C-59.11) fell within the confidence bands. Figure 9a–c show similar tendencies observed in [11] by the NNb model.

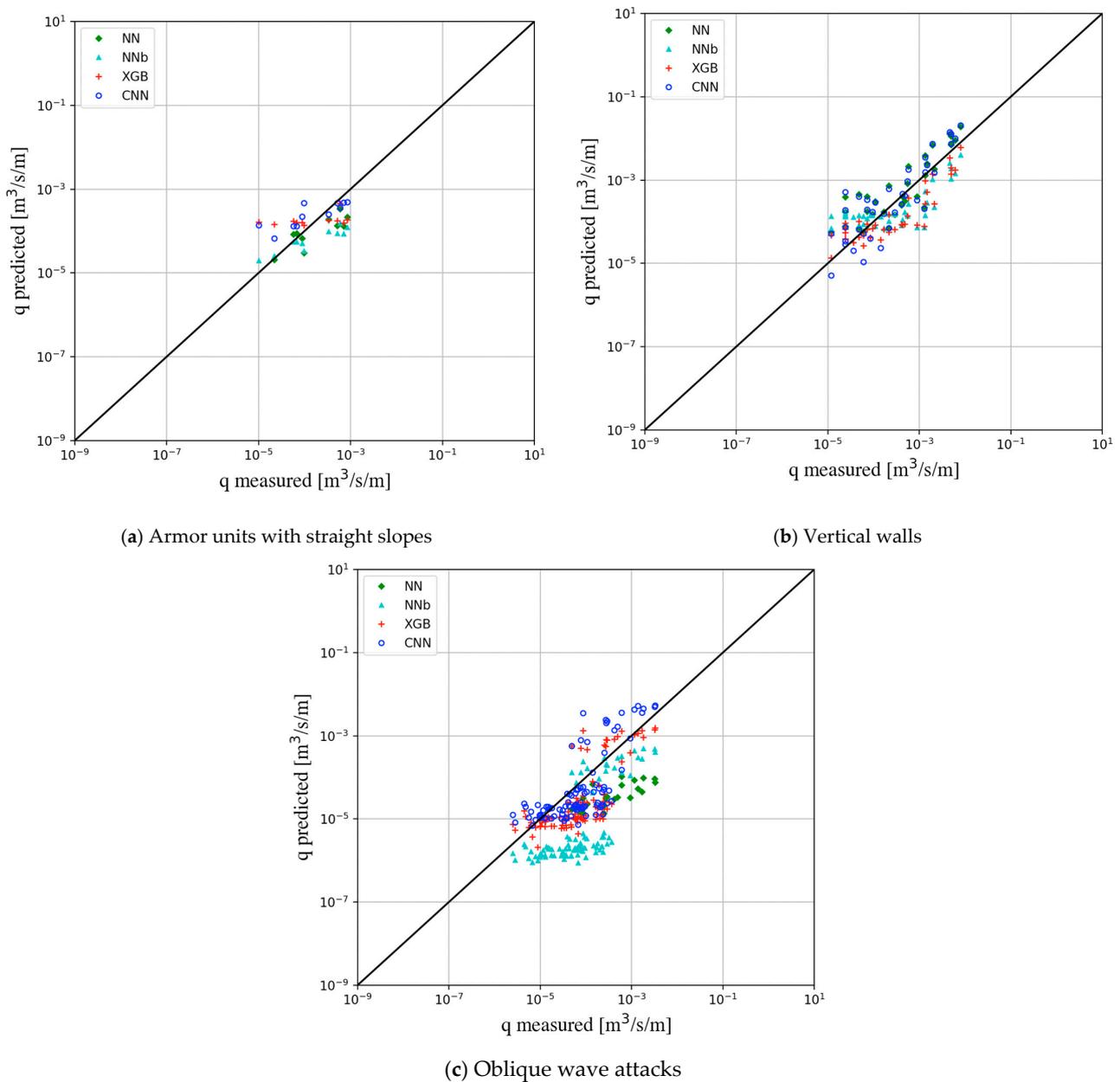


Figure 9. Comparisons of predicted wave overtopping discharges of different models for various coastal structures.

4.4. Application of CNN for Prototype Various Crest Freeboards

In an example scenario, each model was applied to predict the prototype wave overtopping discharge, considering various crest freeboards (R_c) of coastal structures. The predictions were based on the observation labeled as G-12.22 in the prototype situations of the EurOtop database. The input conditions are presented in Table 6. The predictions started with $R_c = 3.34$ m and gradually increased to 7.50 m to observe the trend of the wave overtopping discharge, with a measured observation available at $R_c = 7.36$ m.

Table 6. Conditions for estimating overtopping discharge in example scenario.

Parameter	Value	Parameter	Value	Parameter	Value
$H_{m0,t}$ [m]	1.67	R_c [m]	3.34 to 7.50	D_d [m]	0.60
h_b [m]	1.03	A_c [m]	3.34	D_u [m]	0.00
h_t [m]	3.28	G_c [m]	1.00	γ_{fd}	0.55
h [m]	3.28	B [m]	8.00	γ_{fu}	1.00
$L_{m-1,0,t}$ [m]	59.39	$cot\alpha_d$	1.00	β [°]	14
B_t [m]	0.00	$cot\alpha_u$	0.00		

In Figure 10, the depicted results illustrate the predicted outcomes of relative wave overtopping discharge concerning the relative crest freeboard for four different models: the current CNN, XGB, NN, and NNb models. These results reveal a consistent trend across all predictions, where a lower relative crest freeboard corresponds to a larger relative wave overtopping discharge. Along the projected curve, the predictions generated by the current CNN model closely align with the observed measurements at $R_c = 7.36$ m, specifically at $R_c/H_{m0,t} = 4.41$. In contrast, the predictions produced by the NN, NNb, and XGB models notably underestimate the relative wave overtopping discharge for various relative $R_c/H_{m0,t}$ values.

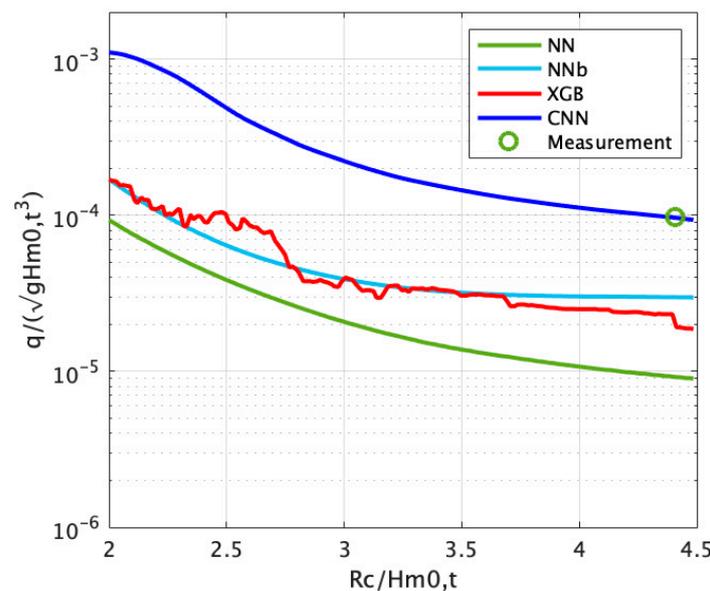


Figure 10. Comparisons of the predicted relative wave overtopping discharge versus relative crest freeboard using different models for prototype situations.

5. Conclusions

In this paper, a novel convolutional neural network (CNN) model with deep learning technologies is introduced for the accurate prediction of wave overtopping discharge at various types of coastal structures, including those with complex geometries and under a wide range of wave conditions. The proposed CNN model incorporates various hidden layers, such as bottleneck residual blocks, which effectively extract features and scale input sequences. For the enhancement of prediction accuracy, hyperparameters of the model are optimized through Bayesian optimization to mitigate overfitting, and the model is trained utilizing the bootstrap resampling technique. The results demonstrate that the proposed CNN model outperforms traditional machine learning models, as evident from comparisons of quantile–quantile plots and root-mean-square errors for estimations on the overtopping database. Notably, the CNN model exhibits remarkable agreement with observations across various structure types and oblique wave attack conditions. The model also demonstrates satisfactory performance for prototype overtopping prediction,

highlighting its value as a valuable tool for coastal engineers involved in the design and maintenance of coastal structures.

Supplementary Materials: Our CNN model can be accessed online at <https://github.com/fcuyttsai/overtoppingEve.git/> (accessed on 1 September 2023).

Author Contributions: Conceptualization, Y.-T.T. and C.-P.T.; Methodology, Y.-T.T. and C.-P.T.; Data curation, Y.-T.T. and C.-P.T.; Validation, Y.-T.T. and C.-P.T.; Formal analysis, Y.-T.T. and C.-P.T.; Visualization, Y.-T.T. and C.-P.T.; Software, Y.-T.T. and C.-P.T.; Writing—original draft, Y.-T.T. and C.-P.T.; Writing—review and editing, Y.-T.T. and C.-P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council of Taiwan for financial support under Grant Nos. MOST 109-2625-M-005-006-MY2, NSTC 111-2221-E-035-044, and MOST 109-2221-E-035-001-MY2.

Data Availability Statement: The authors do not have permission to share data.

Acknowledgments: The authors would like to thank the EurOtop team for providing the overtopping database.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Troch, P.; Geeraerts, J.; Van de Walle, B.; De Rouck, J.; Van Damme, L.; Allsop, W.; Franco, L. Full-scale wave-overtopping measurements on the Zeebrugge rubble mound breakwater. *Coast. Eng.* **2004**, *51*, 609–628. [\[CrossRef\]](#)
2. Briganti, R.; Bellotti, G.; Franco, L.; De Rouck, J.; Geeraerts, J. Field measurements of wave overtopping at the rubble mound breakwater of Rome–Ostia yacht harbour. *Coast. Eng.* **2005**, *52*, 1155–1174. [\[CrossRef\]](#)
3. Cáceres, I.; Sánchez-Arcilla, A.; Zanuttigh, B.; Lamberti, A.; Franco, L. Wave overtopping and induced currents at emergent low crested structures. *Coast. Eng.* **2005**, *52*, 931–947. [\[CrossRef\]](#)
4. van Gent, M.R.A.; van den Boogaard, H.F.P.; Pozueta, B.; Medina, J.R. Neural network modelling of wave overtopping at coastal structures. *Coast. Eng.* **2007**, *54*, 586–593. [\[CrossRef\]](#)
5. Losada, I.J.; Lara, J.L.; Guanche, R.; Gonzalez-Ondina, J.M. Numerical analysis of wave overtopping of rubble mound breakwaters. *Coast. Eng.* **2008**, *55*, 47–62. [\[CrossRef\]](#)
6. Reeve, D.E.; Soliman, A.; Lin, P.Z. Numerical study of combined overflow and wave overtopping over a smooth impermeable seawall. *Coast. Eng.* **2008**, *55*, 155–166. [\[CrossRef\]](#)
7. Verhaeghe, H.; De Rouck, J.; van der Meer, J. Combined classifier–quantifier model: A 2-phases neural model for prediction of wave overtopping at coastal structures. *Coast. Eng.* **2008**, *55*, 357–374. [\[CrossRef\]](#)
8. De Rouck, J.; Verhaeghe, H.; Geeraerts, J. Crest level assessment of coastal structures—General overview. *Coast. Eng.* **2009**, *56*, 99–107. [\[CrossRef\]](#)
9. EurOtop. *Manual on Wave Overtopping of Sea Defences and Related Structures. An Overtopping Manual Largely Based on European Research, but for Worldwide Application*; Van der Meer, J.W.; Allsop, N.W.H.; Bruce, T.; De Rouck, J.; Kortenhaus, A.; Pullen, T.; Schüttrumpf, H.; Troch, P.; Zanuttigh, B.: 2018. Available online: www.overtopping-manual.com (accessed on 1 November 2019).
10. van der Meer, J.W.; Verhaeghe, H.; Steendam, G.J. The new wave overtopping database for coastal structures. *Coast. Eng.* **2009**, *56*, 108–120. [\[CrossRef\]](#)
11. Zanuttigh, B.; Formentin, S.M.; van der Meer, J.W. Prediction of extreme and tolerable wave overtopping discharges through an advanced neural network. *Ocean Eng.* **2016**, *127*, 7–22. [\[CrossRef\]](#)
12. den Bieman, J.P.; van Gent, M.R.A.; van den Boogaard, H.F.P. Wave overtopping predictions using an advanced machine learning technique. *Coast. Eng.* **2021**, *166*, 103830. [\[CrossRef\]](#)
13. Heaton, J. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning. *Genet. Program. Evolvable Mach.* **2018**, *19*, 305–307. [\[CrossRef\]](#)
14. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural. Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
15. Formentin, S.M.; Zanuttigh, B.; van der Meer, J.W. The New EurOtop Neural Network Tool for an Improved Prediction of Wave Overtopping. In *ICE Coasts, Marine Structures and Breakwaters*; ICE Publishing: Leeds, UK, 2017.
16. Formentin, S.M.; Zanuttigh, B.; van der Meer, J.W. A Neural Network Tool for Predicting Wave Reflection, Overtopping and Transmission. *Coast. Eng. J.* **2018**, *59*, 1750006-1–1750006-31. [\[CrossRef\]](#)
17. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1909.11855.
18. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

19. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
20. Ying, X. An Overview of Overfitting and its Solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [[CrossRef](#)]
21. Isabona, J.; Imoize, A.L.; Kim, Y. Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning. *Sensors* **2022**, *22*, 3776. [[CrossRef](#)]
22. Söbester, A.; Leary, S.J.; Keane, A.J. On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. *J. Glob. Optim.* **2005**, *33*, 31–59. [[CrossRef](#)]
23. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
24. Mueller, J.; Jaakkola, T. Principal Differences Analysis: Interpretable Characterization of Differences between Distributions. *arXiv* **2015**, arXiv:1510.08956.
25. Diederik, P.; Kingma, J.B. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.
26. Scikit-Optimize. Sequential Model-Based Optimization in Python. Available online: <https://scikit-optimize.github.io/> (accessed on 1 May 2023).
27. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
28. Geeraerts, J.; Boone, C. *Report on Full Scale Measurements Zeebrugge, 2nd Full Winter Season*; Ghent University: Gent, Belgium, 2004.
29. Franco, L.; Briganti, R.; Bellotti, G. *Report on Full Scale Measurements, Ostia, 2nd Full Winter Season*; Modimar: Rome, Italy, 2004.
30. Pullen, T. *Final Report on Laboratory Measurements, Samphire Hoe*; HR Wallingford: Wallingford, UK, 2004.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.