Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA CIVILE, AMBIENTALE E DEI MATERIALI

Ciclo XXVII

Settore Concorsuale di afferenza: 08/A1

Settore Scientifico disciplinare: ICAR/01

NEURAL NETWORK MODELLING OF THE WAVE-STRUCTURE INTERACTION PROCESSES

Presentata da: **Sara Mizar Formentin**

Coordinatore Dottorato	**Relatore**
Prof. Ing. A. Lamberti	**Prof.ssa Ing. B. Zanuttigh**

Esame finale anno 2015

To my mammy
and
my granny

LIST OF CONTENTS

ACKNOWLEDGMENTS

The European Commission, through FPZ ENV2009-1, Contract 244104 THESEUS project ("Innovative technologies for safer European coasts in a changing climate", www.theseusproject.eu) and the Ministry of Research and University, through the national RITMARE project ("La ricerca italiana per il mare", www.ritmare.it) are gratefully acknowledged for the financial support.

I would like to heartily thank my supervisor, Prof. Eng. Barbara Zanuttigh, who gave me the opportunity to start and carry out this work, and for prompting, supporting and encouraging me from both a professional and a personal point of view.

Prof. Eng. Jentsje W. van der Meer is sincerely acknowledged for sharing his experience with me and for his precious hints that contributed to sustain and address this research. Besides, I am very grateful for providing me with the hard copies of the CLASH database and driving me in the revision and understanding of the tests.

A special thank goes to Dr. Eng. Tonino Liserra, informatics "teacher" and friend.

And finally, thanks to all those who were there for me and with me and know why I did not include them in this list.

SUMMARY (in English)

This thesis presents a new Artificial Neural Network (ANN) able to predict at once the main parameters representative of the wave-structure interaction processes, i.e. the wave overtopping discharge, the wave transmission coefficient and the wave reflection coefficient. The new ANN has been specifically developed in order to provide managers and scientists with a tool that can be efficiently used for design purposes.

The development of this ANN started with the preparation of a new extended and homogeneous database that collects all the available tests reporting at least one of the three parameters, for a total amount of 16'165 data. The variety of structure types and wave attack conditions in the database includes smooth, rock and armour unit slopes, berm breakwaters, vertical walls, low crested structures, oblique wave attacks.

Some of the existing ANNs were compared and improved, leading to the selection of a final ANN, whose architecture was optimized through an in-depth sensitivity analysis to the training parameters of the ANN. Each of the selected 15 input parameters represents a physical aspect of the wave-structure interaction process, describing the wave attack (wave steepness and obliquity, breaking and shoaling factors), the structure geometry (submergence, straight or non-straight slope, with or without berm or toe, presence or not of a crown wall), or the structure type (smooth or covered by an armour layer, with permeable or impermeable core).

The advanced ANN here proposed provides accurate predictions for all the three parameters, and demonstrates to overcome the limits imposed by the traditional formulae and approach adopted so far by some of the existing ANNs. The possibility to adopt just one model to obtain a handy and accurate evaluation of the overall performance of a coastal or harbor structure represents the most important and exportable result of the work.

SOMMARIO (in Italian)

Questa tesi presenta una nuova Rete Neurale Artificiale (RNA) in grado di predire i principali parametri rappresentativi del processo di interazione onda-struttura, ossia la portata di tracimazione ondosa, il coefficiente di trasmissione ondosa e il coefficiente di riflessione ondosa. Tale nuovo modello di RNA è stato sviluppato con lo specifico scopo di fornire un valido strumento dal semplice impiego, che possa essere utilizzato a fini progettuali.

Il primo passo per la realizzazione di questa RNA ha consistito nella raccolta e organizzazione di tutti i dati attualmente disponibili in letteratura, che riportassero il valore di almeno uno dei tre parametri da predire. Complessivamente, sono stati assemblati 16'165 dati in unico database omogeneo, includendo una vasta tipologia di opere, fra cui: strutture lisce e impermeabili, strutture permeabili in massi naturali o rivestite di unità artificiali in cemento, opere con e senza nucleo impermeabile, muri a parete verticale, strutture dalla geometria articolata da berme e/o protezioni al piede, strutture a cresta bassa, condizioni di attacco ondoso obliquo.

L'assetto finale della RNA è stato definito mediante il confronto e il miglioramento di alcune delle reti esistenti, e in seguito a un'approfondita analisi di sensitività ai diversi parametri di calibrazione del modello. Ciascuno dei 15 parametri di ingresso della rete è finalizzato alla rappresentazione di un diverso aspetto dell'unico fenomeno dell'interazione onda-struttura, descrivendo la tipologia di attacco ondoso (ripidità e obliquità delle onde, indici di frangimento e di shoaling), la sezione geometrica (sommergenza, caratteristiche del paramento a mare, quali presenza o assenza di berme e protezioni al piede, presenza o meno di muro di coronamento) e il tipo di struttura (liscia, o rivestita di una mantellata di massi artificiali o naturali, con o senza nucleo impermeabile).

La RNA proposta nell'ambito di tale elaborato produce stime accurate per tutti e tre i parametri ed è in grado di superare i limiti imposti dalle formule tradizionali presenti in letteratura e dalle RNA esistenti, usualmente ottimizzate per la predizione di uno solo dei parametri. La possibilità di utilizzare un unico modello che fornisca una stima accurata, e al tempo stesso rapida, della risposta idraulica di una struttura di difesa costiera o portuale alla sollecitazione ondosa rappresenta il risultato più importante ed esportabile di tale lavoro.

1. INTRODUCTION

1.1 Motivations and background

The assessment of the performance of coastal and harbor structures for design purpose requires the accurate analysis of all the wave-structure interaction processes, which can be essentially described through three quantities: the overtopping discharge, the wave reflection coefficient and the wave transmission coefficient.

Most of the existing formulae and models are targeted to represent one process and are fitted on specific (more or less wide) databases, addressing usually one or few structure types and having therefore a specific (more or less narrow) validity field.

In alternative to traditional techniques Artificial Neural Networks (ANNs) offer flexibility and accuracy. ANNs have been already successfully used in Coastal Engineering in a wide range of applications. This kind of tools are able to deal also with complicated structure geometries and variable wave conditions. Specific ANNs have been developed in the last years for the representation of the main wave-structure processes, and most of them actually proved to be able to overcome some of the limits imposed by the traditional empirical formulae. However, they are still restricted to reproduce only one of the processes involved in the wave-structure interaction (i.e. wave reflection, or wave transmission, etc.).

Notwithstanding this approach, the three phenomena should be considered as different outcomes of the same physical process, and therefore should be investigated contemporarily. The assumption that all the processes are physically correlated implies that a unique set of physically-based parameters can be defined to represent all the phenomena. Then, the development of a predicting method which can represent all the wave-structure interaction processes and which can be implemented in a design support system is supposedly possible.

1.2 Objectives

On the basis of the promising results achieved by the neural networks modelling, the objective of this research is to develop an ANN for the representation of the overall phenomenon of the wave-structure interaction. Ultimate aim of the study is therefore to deliver a tool able to estimate the wave overtopping discharge (q), the wave transmission and the wave reflection coefficients (K_r and K_t) at once, i.e. by means of just one set of input parameters and of the same ANN architecture.

The ANN model is expected to represent a valid and handy tool which can be easily and efficiently employed for the design of coastal and harbour structures.

For this purpose, a robust and versatile architecture will be defined after an accurate and in-depth sensitivity analysis of the several input and internal parameters, while the calibration and the validation of the ANN will be based on a homogeneous and “wide-enough” database, collecting the so-far available experimental tests.

The outcomes of the ANN will be finally compared to the existing tools, in order to illustrate and discuss the field of validity of the model, its potentialities and its skills.

1.3 Structure of the thesis

The research has essentially consisted in a step-by-step work of literature review and optimization processes, which have contributed to the final advanced layout of the ANN. For this reason, the presentation of the work, and in particular of the results of the sensitivity analysis will partially follow the chronological steps and will sometimes refer to preliminary configurations of the optimized ANN (in these cases, it will be specified in the text).

The thesis is organized in the following 7 sections (besides the present introducing one):

- The neural network modelling, from a conceptual and operative point of view, is illustrated in Chapter 2. Aim of this section is to provide the reader with a general idea of this kind of modelling, in order to become confident with the most important and recurrent ANN features.
- Chapter 3 aims to list and present some of the most recent and significant applications of the neural networks in the field of the Ocean and Coastal Engineering. Particular attention will be paid to the three existing networks developed for the single prediction of the wave-structure interaction parameters (q, K_r and K_t). These ANNs will be considered as basis for comparison with the new ANN performance.
- Within Chapter 4, the characteristics and the outline of the newly developed ANN are illustrated. A first part of this section describes the collected database, clarifying its format and the way the data are organized and distributed. Then, the ANN input parameters and architecture are presented, together with the main results of the sensitivity analysis which justify the choice of the final features.
- Chapter 5 shows and discusses the results of the ANN. The prediction of q, K_r and K_t is qualitatively investigated through plots and diagrams which report the comparison among experimental and predicted values and the distribution of the errors computed by the ANN. Furthermore, a quantitative analysis of the results is provided by means of error indexes and computation of confidence bands. As anticipated, the ANN predictions will be compared to the ones derived from the existing ANNs. Besides, a comparison with some of the traditional formulae developed for the prediction of the three parameters is provided.
- Chapter 6 presents the modification introduced in the ANN architecture regarding the specific prediction of q. The revision of the model layout – finalized to optimize the ANN capability to predict the small or null values of q – is subject of ongoing research. However, the developed preliminary methodology is here reported, mainly to introduce the approach we are adopting and provide an idea of the likely final architecture of the ANN.
- Conclusions and steps for further research are drawn in the final Chapter 7.
- References are listed in Chapter 8.

2. MODELLING THROUGH ARTIFICIAL NEURAL NETWORKS

2.1 Introduction

Artificial Neural Networks (ANNs) are computational algorithms belonging to the family of the so-called “machine learning” tools. The conceptual scheme of an ANN is inspired by the human brain and its biological neurons. As the human brain deals with reality by storing the information into the neurons and organizes the knowledge by means of the synapses, i.e. by connecting the neurons each other up to creating a neural network, so the Artificial Neural Networks “learn” and reproduce physical input-output patterns by means of artificial neurons and artificial connections among them.

ANNs actually consist in “black-box” numerical models, which essentially analyze and re-elaborate experimental data, “learning” and reproducing the cause-effect relationships but not investigating them. ANNs focus on the input-outputs links, completely disregarding the study of the governing physical process and without attempting to introduce mathematical formulations.

The construction of an ANN starts with the identification of the governing parameters of the process to be modeled, or the most relevant factors affecting the output to be predicted. The selected parameters will compose the input set of the ANN. Often it is not possible to know since the beginning the correct number and combination of the input parameters, so the input set is generally defined after a sensitivity analysis to the governing factors.

The adoption of an ANN may be particularly useful and recommended in case of complex empirical phenomena whose proper description implies the introduction of several physical parameters, but the first and utmost requirement for the development of an ANN is the availability of a homogeneous and reliable experimental database.

Therefore, these kind of tools are also known as “data driven models”, since the operating principle is based on a learning process on experimental (or artificial) data finalized to reproduce the existing relationships among the data themselves. From this viewpoint, the experimental data are not only a calibration tool, but represent the starting point and the key element within the “building” process of the model. Consequently, the reliability and the accuracy of the ANN outcomes strictly depend on the precision and reliability of the available experimental data.

As stated in Chapter 1, the main goal of this study is to develop and provide an ANN that can predict the wave overtopping discharge, the wave reflection and the wave transmission coefficients at once, trying to improve the results compared to previous prediction formulae and ANNs. Aim of this Chapter is therefore to describe the main features and the architecture of an ANN, in order to briefly explain or at least give an idea of what ANNs are. For this purposes, Paragraphs 2.2 and 2.3 present and illustrate an ANN working principle, respectively following a conceptual and an operative approach. Finally, Paragraph 2.4 resumes the principal concepts and draws some conclusions.

These introductive Paragraphs are mainly based on the Help Tool provided by the software *Matlab* (http://it.mathworks.com/products/neural-network/) and were partially derived from Formentin and Zanuttigh (2012).

2.2 Conceptual layout of an ANN

Artificial Neural Networks are composed of simple elements operating in parallel. Biological nervous systems inspire these elements. As in nature, the connections between elements largely determine the network function. From this concept, the idea to "train" an Artificial Neural Network to perform a particular function by adjusting the values of the connections (weights) between elements has born.

Typically, ANNs are adjusted, or trained, so that a particular input leads to a specific target output. Fig. 2.1 illustrates such a situation: there, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Many such input/target pairs are generally needed to train a network. From a theoretical point of view, the network "training" and the weights adjustment go on until the predicted outputs and the targets exactly coincide. In practice, the training is arrested when a user-defined reasonable value of tolerable error is pursued or when a specific criterion is satisfied.

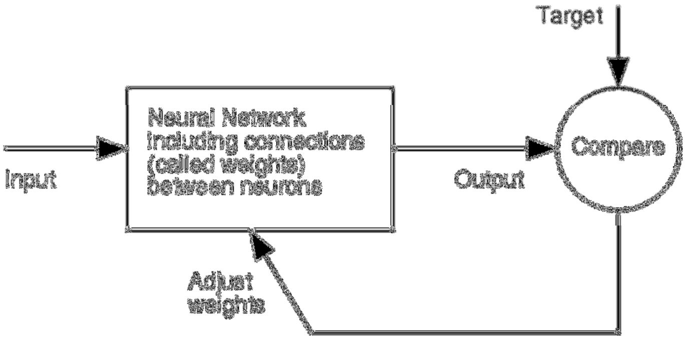

Target
Neural Network including connections (called weights) between neurons
Input
Output
Compare
Adjust weights

Figure 2.1 – Logical layout of an Artificial Neural Network

An ANN may be characterized by several different architectures, but the most developed and employed models are the so-called "multi-layer" (ML) networks. A ML is organized in sequentially-disposed "layers": the input parameters set may be considered as the first layer (it is preferable to consider it simply as the "input vector", being the layers properly belonging to the ANN model and not to the user-defined input set), while the output parameters set is actually the last layer of a ML network. Between the input vector and the output layer may be present one or more layer, the so-called "hidden layers".

Each layer is composed by one or more neurons, the designated elements for the data elaboration that work in parallel (within a same layer). By seeing at the input vector as an improper layer, its elements – i.e. each physical parameter which is supposed to be involved in the empirical phenomenon to be modeled – may be considered as "input neurons" (hereafter, it will be more properly referred as "input element"). The output layer includes as many "output neurons" as the output parameters to be predicted through the ANN. The hidden layer(s) may contain a variable number of "hidden neurons", a number which cannot be defined *a priori* but which need to be established by means of an iterative process of calibration.

Figure 2.2 reports an example of an ANN architecture by displaying the different layers and the respective neurons.

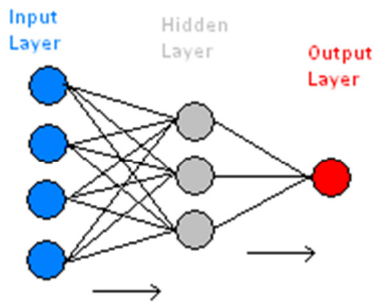


Figure 2.2 – Example of a multi-layer ANN logical architecture.

Generally, increasing the number of hidden neurons causes an increase of the ANN performance. Indeed, each neuron corresponds to a degree of freedom of the model, and therefore an increased capability of represent the under-laying physical process. However, the increase of hidden neurons is upper-limited: too many neurons generate "noisy fluctuations" in the learning phase, due to the excessive complexity of the architecture, and the ANN performance does not improve anymore. The ANN becomes "over-trained".

An ANN is only one-direction working, i.e. it can process data only from input to output. The information is passed from one layer to the subsequent one trough the connections among neurons. Each neuron is connected to all the neurons belonging to the following layer, but not to the neurons belonging to the same layer: for example, considering a layer composed by n neurons and a following one composed by m neurons, the resulting number of connections between the two layers is n x m.

Each neuron j belonging to an inner (or output) layer receives as input (or "activation value") the weighted sum

$$\sum_{i=1}^{n} (X_i W_{ij})$$

Eq. 2.1

of each output $X_{1,2,..,n}$ value belonging to the previous layer. The activation value is then processed by a "transformation function" (or "activation function") which transforms the activation value into a output, ready to be passed to the following layer. The transformation function is activated when the activation value equals or exceeds a threshold value. The threshold value is essentially governed by a "bias" value B_j that is summed to the activation value, in order to shift the activation value and ensure the correct working of the neuron. Then, the "complete" activation value passed to a neuron j is:

$$A_j = \sum_{i=1}^{n} (X_i W_{ij} + B_j),$$

Eq. 2.2

where n is the number of neurons belonging to the preceding layer and W_{ij} is a coefficient associated to the connection among the element X_i and the neuron j.

During the "learning" process, the ANN associates a weight to each connection: the weight W_{ij} is a numerical coefficient whose value is set and iteratively adjusted by specific algorithms implemented by the ANN during the learning process itself. Depending on the value of the attributed weight, the relative importance of a connection varies, defining the relevance of the information associated to that connection. Of course, the higher the weights values, the higher the estimated importance of a corresponding connection and therefore the higher relevance of that specific input-output correlation.

The weights of the several connections may be interpreted as the model unknown parameters to be calibrated: the learning phase of the ANN actually consists in the definition of the weights values in order to best reproduce the input-output pattern provided by the experimental data and targets.

The modifications to the weights values are driven by an iterative process of minimization of the error among the ANN predictions and the targets. At each step, the ANN performs a guess solution (i.e. a guess pattern of weights) to be compared to the real targets, computes the committed error and, through specific algorithms of error minimization, provides the ANN a correction to the previous guess solution, in order to start a new iteration. This overall process is the "training" phase, which comprehends the learning phase and implements several algorithms besides the minimization procedure of the error. For example, the training includes algorithms:

- to speed-up (if possible) the minimization process;
- to stop the learning process when no (or small) improvement is achieved after a fixed number of iterations;
- to extend the ANN capability of generalization, i.e. of correctly dealing with new data which do not exactly fall within the range of the target values employed in the training.

2.2.1 Training an ANN

The "training" phase comprehends and partially corresponds to the more specific phase of "learning". During these phases, the ANN actually "learns" from the experimental target values how to adjust the weights in order to re-elaborate the input-output patterns. The ANN training, is based on the so-called "*Error Back Propagation Rule*", and follows these steps:

- the overall available database is generally split into three parts, the "training set", the "testing set" and the "validating set". The first one contains the data employed to learn and reproduce the input-output patterns. The second one is adopted to compare, at each iteration, the ANNs outputs and the targets and computing the committed error. The last one, which is used to test the ANN performance as well as the testing set, is adopted to stop the training when certain conditions are satisfied (this issue will be discussed in detail in the next Paragraph 2.2.2).
- the first layer of the network, on the basis of the inputs (belonging to the training set) creates a first guess of values for the weights and the bias, performs the transfer function and passes to the following layer, up to the output one;
- the data belonging to the testing set are passed to the temporarily-trained network (i.e. the resulting network trained after only the first iteration), which processes the new inputs (the testing set must be absolutely separated from the training set) and performs the outputs;
- the predicted outputs are compared to the corresponding target values of the testing set, and the error is computed; the error propagates within the ANN according to the *back-propagation rule* (i.e. from output to input, backwards);
- based on the computed error (generally, the mean square error, *mse*, or its variations), the weights and the bias are corrected and the cycle re-starts with a new iteration;
- the iterations go on until the error equals zero or (much more common in practical cases) no "significant" improvements between an iteration and the following one are pursued.

The re-arrangement of weights values, iteration after iteration, is defined by a minimization function of the error, considered as a cost or performance function (*mse*). The performance function allows evaluating how much the predictions are far from the experimental values, and therefore how much the model is far from the optimal trained condition.

Much attention must be paid to the choice of the training set and the testing set: the separation of the database within these datasets may significantly affect the ANN performance. Indeed, defining these sets implies choosing which tests are to be employed to "train" the ANN and which ones to verify its performance. It is pretty clear that if training and testing sets contain non-homogenous tests, i.e. if the training set does not include the complete range of variability of the input parameters, the ANN performance will result unsatisfactory, since the ANN is asked to deal with completely out-of-range data within the testing set.

A best practice is to vary the partition into the three sets and re-train the ANN several times. For this purpose – which allows also carrying out a sensitivity analysis to the distribution of experimental data – the so-called "Resampling Techniques" exist. The adoption of these techniques is strictly correlated to the utility and the meaning of the validating set, and will be discussed in the next section 2.2.2.

2.2.2 Validating set and improving generalization

One of the most important issues correlated to the actual performance of an ANN is represented by its capability of generalization, i.e. of overcoming the limits of the range of training tests. An ANN is said to be "over-trained" when it is able to reproduce very well its training data, but is not able to predict with sufficient accuracy beyond the training ranges.

This generally occurs when a "too-large" number of hidden neurons has been applied and the architecture of the ANN is "too much focused" on replicate the trend of the training tests and does not learn the "general rule". For example, Fig. 2.3 displays the response of a network trained to approximate a noisy sine function. The underlying sine function is shown by the dotted line, the noisy measurements are given by the '+' symbols, and the ANN response is given by the solid line. This network clearly over-fits the data and cannot generalize well: it is "over-trained".

Reducing the number of hidden neurons may be a good strategy to prevent or the risk of over-training, but often – especially if wide databases are employed – a pretty-large number of hidden neurons is anyway requested to obtain satisfactory results. In these cases, the adoption of specific techniques to improve generalization is recommended.

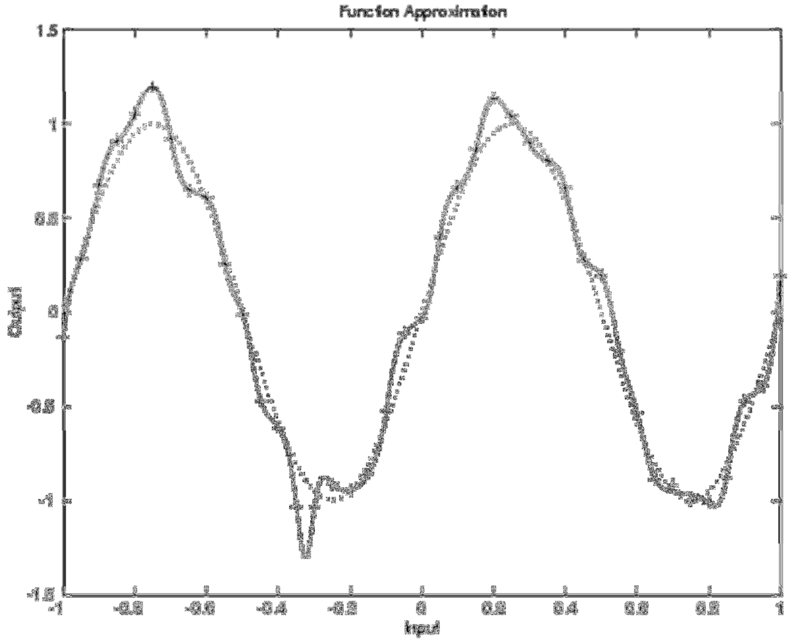


Figure 2.3 – Response of a network trained to approximate a noisy sine function. The underlying sine function is shown by the dotted line, the noisy measurements are given by the '+' symbols, and the ANN response is given by the solid line. Example of an over-fitted network.

The most common technique is the so-called "early-stopping" of the training process. This methodology involves the splitting of the database into the three datasets anticipated in the previous Paragraph 2.2.1 and, into specific, the adoption of the validating set. The validating set, as well as the training set, is an independent set of data adopted to test the ANN performance at each step of the iterative process of training.

Differently from the testing set, the performance function computed at each iteration to evaluate the ANN error relating to the validating set does not contribute to modify the weights values and therefore does not directly take part to the learning process. The validation performance function is employed to "early" stop the training procedure before the achievement of the expected performance: the stopping is imposed when for several iterations consecutively the performance function on the validating set is not reduced, even if the training error continues to decay. In other terms, the "early-stopping" technique interrupts the training process before the "optimum", leading to a slightly worse performance, but ensuring a greater capability of generalization. Therefore, the validating set helps to test the ANN capability of generalization, i.e. the capability to predict values not belonging to the training set.

However, the introduction of the "early-stopping" may cause a decrease of the ANN performance, and the contextual "improved generalization" may not sufficiently balance the decay. When the "early-stopping" is not suitable to be implemented, other techniques to improve the ANN generalization have been developed and are available. The so-called "bootstrapping" (described in the next Paragraph 2.2.3), belonging to the category of the resampling techniques, can be considered as a methodology for the improvement of the generalization which avoid to early stop the training and splitting the database into three parts.

2.2.3 Resampling techniques: the bootstrap

The "resampling techniques" are methodologies generally adopted to assess the performance and the reliability of numerical or statistical models. As the terms suggest, they consist in the repeated re-sampling of a database aimed to create several different subsets of the database itself and test the model sensitivity to the different distributions of the data.

In the specific case of neural networks, resampling techniques are adopted to resample the training database, in order to re-train more times the same ANN, each time against a differently assorted database, and test the derived performance.

Two of the most common resampling techniques are the "jackknifing" and the "bootstrapping". The bootstrapping – more frequently adopted for ANN applications in coastal engineering filed (Panizzo and Briganti, 2007; Van Gent et al., 2007; Verhaeghe et al., 2008; Zanuttigh et al., 2013) – consists in several (*N*) resamplings with replacement of the data to be selected for the training of the ANN. For each run of the ANN, one of the *N* bootstrapped databases is used for the training. The size of each bootstrapped database equals the original one, but the included data are differently assorted, since each selected test is randomly selected with replacement.

The random selection could be driven by weight factors WF associated to each test: the higher the weight factor, the higher the probability for a test to be selected. Each bootstrapped database may therefore include the same tests more than once, while some tests may never appear.

The bootstrap resampling of the database is principally adopted to assess the performance of the ANN. Each differently trained ANN yields to differently evaluated output parameters, and the ensemble of the predicted outputs can be considered as a stochastic variable and therefore used to derive average indexes of performance and standard deviations. Furthermore, if the number of resamples is large-enough to be statistically significant, it is possible to calculate the quantiles of the distribution and derive the confidence intervals. A paramount aspect is that a mean prediction is not only more significant from a statistical view-point, but is also more accurate since it adopt the commitment of several randomly-trained ANNs.

The adoption of the bootstrap resampling of the database allows also assessing the capability of generalization of an ANN avoiding the employment of specific methodologies, such as the early-stopping (Verhaeghe, 2005). In fact, an ANN trained many times, each time on a randomly different (bootstrapped) database, produces "average" predictions and relative indexes of performance. The risk of "over-fitting" a same training database is then bypassed, and there is no need to "early stop" the training.

2.3 Neural Network architecture and structures

Concerning the ANN architectures, two of the most common ones are the "perceptron" and the feedforward-backpropagation models. A perceptron is a quite simple one-layer ANN, more suitable for problems in pattern classification. They are fast and reliable networks, but can implement only linear functions: in order to deal with more complex problems, more sophisticate architectures, such us the feedforward-backpropagation networks, are required.

A feedforward network is a multi-layer ANN, where one-way connections only are allowed (i.e., neurons may be connected only from one layer to the following one, and not *vice versa*, and no connections among neurons belonging to the same layer exist). Normally, networks working with backpropagation algorithms (see Paragraph 2.3.2) present a multi-layer feed-forward architecture.

An ANN model is essentially characterized by:

- its architecture, i.e. the number of layers and the number of neurons within each layer; usually, ANNs are multi-layer structures;
- the way the neurons are connected each other and the functions adopted to "pass" the information from one layer to the following layer;
- the training algorithm, which comprehend the learning one and several other functions to optimize the training of the model itself.

In the following Paragraphs, the most important and representative elements and concepts characterizing a feedforward-backpropagation network will be presented and more in depth analyzed, in order to better comprehend the nature of an ANN model and to become confident with its peculiar structures and functions.

2.3.1 Neuron model

The simplest example of an artificial neuron is subjected to a single scalar input and no bias appears (see Fig. 2.4).

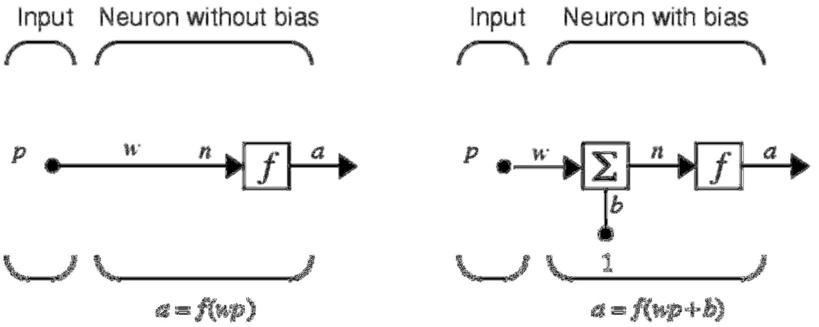


Figure 2.4 – Logical scheme of a neuron without bias (to the left) and with bias (to the right).

The scalar input *p* is transmitted to the neuron through a connection that multiplies its strength by the scalar weight *w*, forming the product *wp*, again a scalar. In this case, the weighted input *wp* is the only argument of the transfer function *f*, which produces the scalar output *a*.

The neuron on the right of Fig. 2.4 has a scalar "bias", *b*. A bias can be viewed as simply element added to the product *wp* in order to shift the function *f* to the left by an amount *b*. The bias is much like a weight, except that it has a constant input of 1 (instead of generically *p*). It is worthy to remark that a bias effectively modifies the input vector of an ANN, but it is not an input element itself. Similarly to the other weights, a bias is created and adjusted by the ANN during the learning process, generally it has not to be created by the user, except than in very specific conditions.

A neuron may also receive an input composed by *R* elements (see Fig. 2.5). The structure does not change: the scalar input *p* becomes an *R*-elements array ***p*** and its individual element inputs p_1, p_2, …, p_R are multiplied by weights w_1, w_2, …, w_R and the weighted values are fed to the summing junction. Their sum is simply ***Wp***, the dot product of the (single row) matrix

W and the vector p. The neuron has a bias *b*, which is summed with the weighted inputs to form the network input *n*=***Wp***+*b*. This sum, the scalar *n*, is the argument of the transfer function *f*, which will perform the scalar output *a*: *f*(*n*)=*a*.

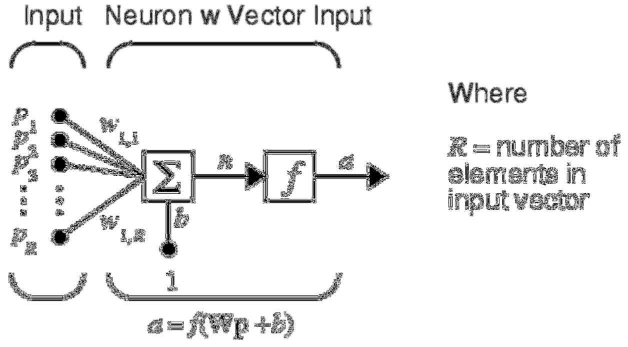


Figure 2.5 – Logical scheme of a neuron subjected to a vector input

Figure 2.5 reports a very detailed layout of a single neuron. When considering networks with many neurons, and perhaps layers of many neurons, this degree of detail should lead to the loss of the main concepts. Thus, the it is possible to substitute this layout with an abbreviated notation for an individual neuron. This notation, which will be used later in circuits of multiple neurons, is shown in Fig. 2.6.

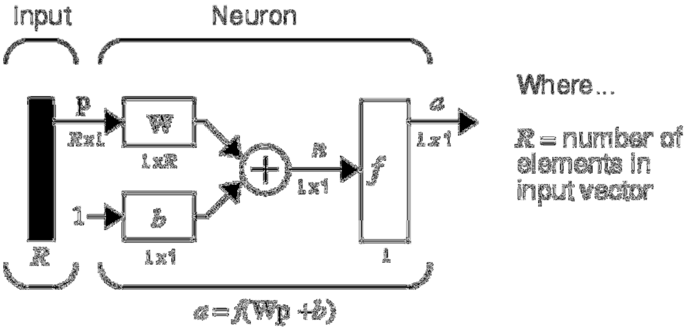


Figure 2.6 – Synthetic logical scheme of a neuron subjected to a vector input.

In Fig. 2.6, the solid dark vertical bar at the left represents the input vector ***p***. The dimensions of ***p*** are shown below the symbol ***p*** in the figure as *R*x1. (Note that a capital letter, such as *R* in the previous sentence, is used when referring to the size of a vector.) Thus, ***p*** is a vector of *R* input elements. These inputs post-multiply the single-row, *R*-column matrix ***W***. As before, a constant 1 enters the neuron as an input and is multiplied by a scalar bias *b*. The net input to the transfer function *f* is *n*. This sum is passed to the transfer function *f* to get the neuron's output *a*. Please, note that if there were more than one neuron, the network output would be a vector. Each time this abbreviated network notation is used, the sizes of the matrices are shown just below their matrix variable names. This notation will allow understanding the architectures and following the matrix mathematics associated with them.

A "layer" of a network is defined in Fig. 2.6. A layer therefore includes the combination of the weights, the multiplication and summing operation (here realized as a vector product ***Wp***), the bias *b*, and the transfer function *f*. The array of inputs, vector ***p***, is not included in or properly called a layer. A network may be composed by two or more neurons in a layer, and more than one layer. Let us first consider a single layer of more neurons: a one-layer network with *R* input elements and *S* neurons is shown in Fig. 2.7.

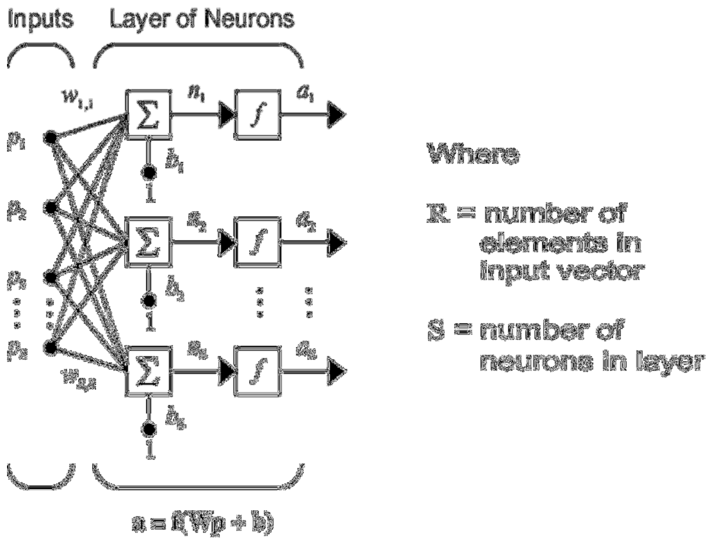


Figure 2.7 – Logical scheme of a layer of neurons subjected to a vector input.

In such a network, each element of the input vector ***p*** is connected to each neuron through the weight matrix ***W***. The *i*-th neuron presents a sum that gathers its weighted inputs and bias to form its own scalar output *n*(*i*). The various *n*(*i*) taken together form an *S*-element input vector *n*. Finally, the neuron layer outputs form a column vector ***a***. The expression for ***a*** is shown at the bottom of the figure: ***a***=*f*(***Wp***+***b***).

The input vector elements enter the network through the weight matrix ***W:***

$$\mathbf{W}=\begin{bmatrix} w_{11} & \cdots & w_{1R} \\ \vdots & \ddots & \vdots \\ w_{S1} & \cdots & w_{SR} \end{bmatrix}$$

Eq. 2.3

The row indices on the elements of matrix ***W*** indicate the destination neuron of the weight, and the column indices indicate which source is the input for that weight. Thus, the indices in $w_{1,2}$ say that the strength of the signal from the second input element to the first (and only) neuron is $w_{1,2}$.

It is worthy to remark that it is common that the number of inputs to a layer is different from the number of neurons (i.e., *R* is not necessarily equal to *S*). A layer is not constrained to have the number of its inputs equal to the number of its neurons. It is possible to create a single (composite) layer of neurons having different transfer functions simply by putting two of the networks shown in Fig. 2.7 in parallel. Both networks would have the same inputs, and each network would create some of the outputs.

The *S* neuron *R* input one-layer network also can be drawn in abbreviated notation (see Fig. 2.8). Here, ***p*** is an *R* length input vector, ***W*** is an *SxR* matrix, and ***a*** and ***b*** are *S*-length vectors. As defined previously, the neuron layer includes the weight matrix, the multiplication operations, the bias vector ***b***, the sum, and the transfer function boxes.

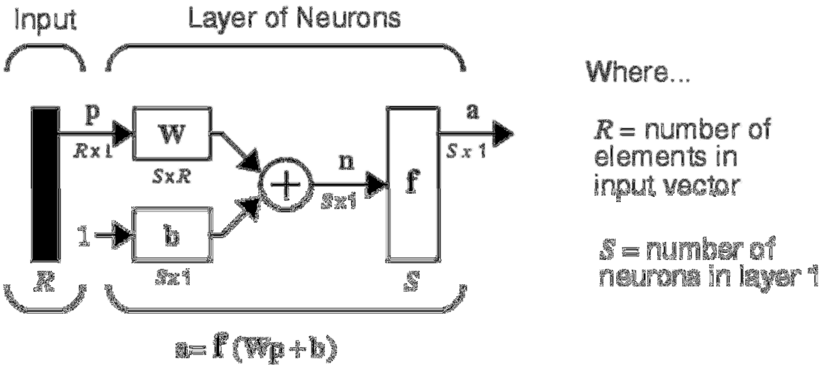

Input
Layer of Neurons
Where...
p
Rx1
W
SxR
n
Sx1
f
a
Sx1
1
b
Sx1
R
S
a= f (Wp+b)
R = number of elements in input vector
S = number of neurons in layer 1

Figure 2.8 – Synthetic logical scheme of a layer of neurons subjected to a vector input.

In order to describe networks having multiple layers, the notation must be extended. Specifically, it needs to make a distinction between weight matrices that are connected to inputs and weight matrices that are connected between layers. It also needs to identify the source and destination for the weight matrices. We will call the weight matrices connected to inputs "Input Weights" (*IW*); we will call weight matrices coming from the outputs of a layer "Layer Weights" (*LW*). Further, superscripts are used to identify the source (second index) and the destination (first index) for the various weights and other elements of the network. To illustrate both these new adopted terminology and symbols, the one-layer multiple input network shown in Fig. 2.8 is re-drawn in abbreviated form in Fig. 2.9.

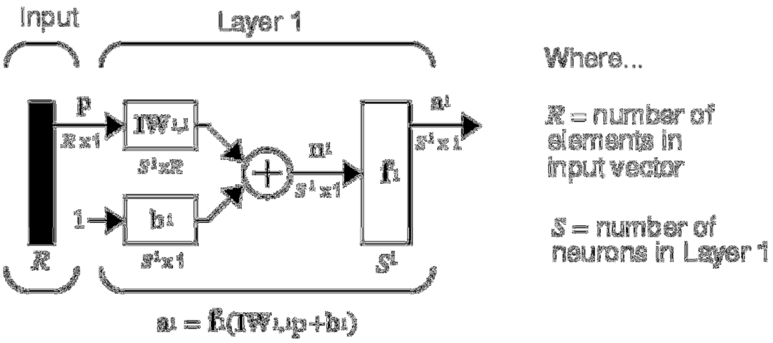


Figure 2.9 – Synthetic logical scheme of a layer of neurons subjected to a vector input belonging to a multi-layer network.

As it can be detected from Fig. 2.9, the weight matrix connected to the input vector ***p*** is labeled as an Input Weight matrix ($\boldsymbol{IW}^{1,1}$) having a source 1 (second index) and a destination 1 (first index). Elements of layer 1, such as its bias, net input, and output have a superscript 1 to say that they are associated with the first layer.

Multiple Layers of Neurons uses Layer Weight (***LW***) matrices as well as input weight (***IW***) matrices.

In the most general case, a network can have several layers. Each layer has a weight matrix ***W***, a bias vector ***b***, and an output vector ***a***. To distinguish between the weight matrices, output vectors, etc., for each of these layers in the figures, the number of the layer is appended as a superscript to the variable of interest. Such a layer notation can be appreciated in the three-layer network shown in both Fig. 2.9, and in the equations at the bottom of the figure. This network has R^1 inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input 1 is fed to the bias for each neuron. The outputs of each intermediate layer are the inputs to the following layer: thus, layer 2 can be analyzed as a one-layer network with S^1 inputs, S^2 neurons, and an S^2x S^1 weight matrix $\boldsymbol{W^2}$. The input to layer 2 is $\boldsymbol{a^1}$; the output is $\boldsymbol{a^2}$.

The layers of a multilayer network play different roles. A layer that produces the network output is called an "output layer". All other layers are called "hidden layers".

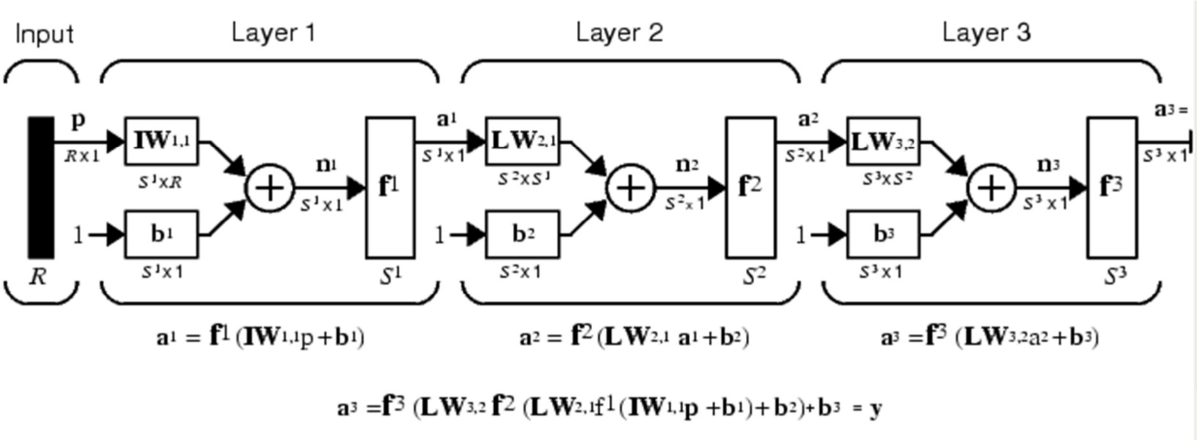


Figure 2.10 – Synthetic logical scheme of multi-layer network.

The three-layer network shown in Fig. 2.10 has definitely one output layer (layer 3) and two hidden layers (layer 1 and layer 2). In this case, it is assumed that the output of the third layer, $\boldsymbol{a^3}$, is the network output of interest, and this output is labeled as $\boldsymbol{y}$. This notation will be used to specify the output of multilayer networks.

Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. This kind of two-layer network is used extensively in Backpropagation (see Paragraph 2.3.5).

2.3.2 Transfer functions

Within Paragraph 2.3, the term "transfer function" has been used (some authors adopt the term "activation function"). A "transfer function" biologically corresponds to the firing of a neuron in case the weighted sum of inputs exceeds a certain threshold value (Verhaeghe, 2005). By referring to the simplest case of ANN model (see Fig. 2.4), a transfer function f receives as input the weighted sum ($wp+b$) and produces the output a:

$$f(wp+b)=a$$

Eq. 2.4

In case of vector inputs, layer composed by more neurons and vector bias, the transfer function keeps the form of Eq. 2.4, and the only changes are related to the arguments of the function (which, respectively, becomes a matrix $\boldsymbol{W}$ and two vectors $\boldsymbol{p}$ and $\boldsymbol{b}$) and to the output $\boldsymbol{a}$ (which becomes a vector). Therefore, a transfer function actually processes an input information and produces an output according to a specific analytical function.

Many transfer functions exist, in order to fit several different input-output experimental patterns. For example, a simple classification problem may be solved by adopting a Boolean-type transfer function (i.e. a function that produces only 0 and 1 outputs); then, according to the increasing complexity of the problem, linear, sigmoid, logarithm, etc. transfer functions are available. Then, the "optimal" transfer function depends on the physical process to be represented, but it is definitely supposed to be chosen after a sensitivity analysis to the several options.

A complete list of the transfer functions collected in the Neural Network Toolbox of *Matlab* is available in the software reference pages. In this work, three of the most commonly used functions are briefly described hereafter (please, note that *Matlab* terms are adopted).

1. "Hard-limit Transfer Function": this function limits the output of the neuron to either 0, if the net input argument *n* is less than 0, or 1, if *n* is greater than or equal to 0. This function is principally used in Perceptron Networks to create neurons that make classification decisions. Fig. 2.11 reports a conceptual layout of this function: as it can be observed, a discontinuity "square profile" form represents its logical operation.

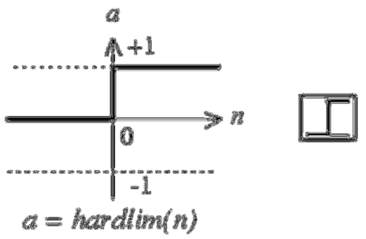


Figure 2.11 – Hard Limit Transfer Function conceptual layout (*Matlab* symbology is adopted).

2. "Linear Transfer Function": this function transforms the input *n* (which can assume any real value) into an output included within the range [-1 ; +1]: if *n* is lower than 0, the output will be negative as well, and included between -1 and 0, if *n* is greater than 0 the output will be positive and lower than 1; if *n* equals 0, the output will be 0 as well. Neurons of this type are used as linear approximators in Linear Filters. As shown, in Fig. 2.12, this function is represented by a continuous straight line (actually a linear function).

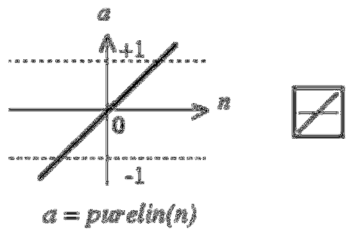


Figure 2.12 – Linear Transfer Function conceptual layout (*Matlab* symbology is adopted).

3. "Log-Sigmoid Transfer Function": the sigmoid transfer function shown to the left of Fig. 2.13 takes the input, which can have any value between plus and minus infinity, and squashes the output into the range 0 to 1. This transfer function is commonly used in backpropagation networks (see Paragraph 2.3.5), in part because it is differentiable.

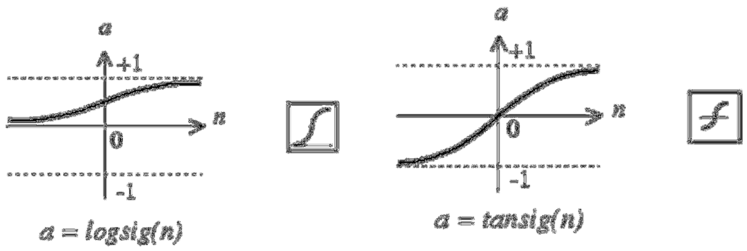


Figure 2.13 – Log-Sigmoid (to the left) and Hyperbolic Tangent Sigmoid (to the right) Transfer Functions conceptual layout (*Matlab* symbology is adopted).

An often-employed alternative function to the log-sigmoid is the Hyperbolic Tangent Sigmoid Transfer Function (Fig. 2.13, right): it still accepts inputs spacing between minus and plus infinity, but squashes them in the continuous range [-1 ; +1]. It is therefore particularly useful when negative output values are expected.

2.3.3 Input data structures

This Paragraph presents how the format of input data structures can be applied to a network and how this issue affects the simulation of networks.

There are two basic types of input vectors: those that occur "concurrently" (i.e. at the same time, or in no particular time sequence), and those that occur "sequentially" in time. For concurrent vectors, the order is not important, and if there were a number of networks running in parallel, you could present one input vector to each of the networks. For sequential vectors, the order in which the vectors appear is important.

The simplest situation for simulating a network occurs when the network to be simulated is static (has no feedback or delays). In this case, there is no need to be concerned about whether or not the input vectors occur in a particular time sequence, so the inputs can be treated as concurrent. In addition, the problem is made even simpler by assuming that the network has only one input vector. A single matrix of concurrent vectors is presented to the network, and the network produces a single matrix of concurrent vectors as output. The result would be the same if there were four networks operating in parallel and each network received one of the input vectors and produced one of the outputs. The ordering of the input vectors is not important, because they do not interact with each other.

When a network contains delays, the input to the network would normally be a sequence of input vectors that occur in a certain time order. Subjected to this type of inputs, the network produces a cell array containing a sequence of outputs. The order of the inputs is important. In this case, the current output is obtained by multiplying the current input by 1 and the preceding input by 2 and summing the result. If the user were to change the order of the inputs, the numbers obtained in the output would change. It the same inputs were presented as a set of concurrent inputs instead of a sequence of inputs, a completely different response would be obtained. However, it is not clear why a user should present concurrent inputs with a dynamic network: it would be as if each input were applied concurrently to a separate parallel network.

Finally, it is also possible to apply sequential inputs to static networks. It does not change the simulated response of the network, but it can affect the way in which the network is trained.

2.3.4 Training styles

As stated in the previous Paragraph 2.3.3, for a static network, the simulation produces the same outputs whether the inputs are presented as a matrix of concurrent vectors or as a cell array of sequential vectors. However, this is not true when training the network. If the inputs are presented as a cell array of sequential vectors, then the weights are updated as each input is presented ("Incremental Training"), while if the inputs are presented as a matrix of concurrent vectors, then the weights are updated only after all inputs are presented ("Batch Training").

The Incremental Training can be applied to both static and dynamic networks, although it is more commonly used with dynamic networks, such as adaptive filters.

In addition, the Batch Training, in which weights and biases are only updated after all the inputs and targets are presented, can be applied to both static and dynamic networks.

Training static networks is relatively straightforward. The format of the input determines the method of training. If the inputs are passed as a sequence, then the network is trained in incremental mode. If the inputs are passed as concurrent vectors, then batch mode training is used.

With dynamic networks, batch mode training is typically done.

2.3.5 Training algorithms

Once the network weights and biases are initialized, the network is ready for training. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function.

This Paragraph aims to describe the main training algorithms for feedforward networks. All these algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called backpropagation, which involves performing computations backward through the network. The backpropagation computation is derived using the chain rule of calculus and is described in Chapter 11 of Hagan et al. (1996). This book provides a clear and detailed survey of basic neural network architectures and learning rules. It emphasizes mathematical analysis of networks, methods of training networks, and application of networks to practical engineering problems.

A feedforward-backpropagation network is potentially able to approximate any kind of physical process, but it strictly requires the adoption of differentiable functions as transfer functions (see Paragraph 2.3.2). Furthermore, the convergence of a backpropagation network is not theoretically guaranteed, differently from the perceptron networks, which always converge for the problems they can solve.

The potential non-convergence is caused by the complexity of the error surface: while the perceptrons are linear networks, a multi-layer ANN deals with highly non-linear functions, and the corresponding error surfaces may be characterized by several local minima. Since the gradient descent is computed along these surfaces, it may happen that the minimization process falls "trapped" within a local minimum instead of reaching the absolute one. The applied initial and boundary conditions may lead to this risk, and the effect of a local minimum may be more or less affecting depending on the proximity of the local minimum itself to the global minimum and to the tolerance accepted.

Therefore, a good practice to ensure the accuracy of the ANN predictions is represented by the repeated re-initialization and re-training of the ANN (see also the resampling techniques, Paragraph 2.2.3).

Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs. There are two features of *Matlab Neural Network Toolbox* software that are designed to improve network generalization: regularization and early-stopping. These features and their use have been discussed in Paragraph 2.2.2.

The basic backpropagation training algorithm, in which the weights are moved in the direction of the negative gradient, is described in the next Paragraph 2.3.5.1. Two later Paragraphs (2.3.5.2 and 2.3.5.3) describe additional complex algorithms that increase the speed of convergence.

2.3.5.1 Basic Feedforward-Backpropagation

Backpropagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule (see Duda and Hart, 1973), in which the network weights and biases are “moved” along the negative of the gradient of the performance function, which is the “direction” along which the minimization of the performance function is fastest. The term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. A single iteration of the basic backpropagation learning algorithm may be expressed as follows:

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-a_k\boldsymbol{g}_k, \text{ where:} \qquad \textbf{Eq. 2.5}$$

- $\boldsymbol{x}_k$ is the current array of weights and biases;
- $\boldsymbol{g}_k$ is the current gradient;
- a_k is the “learning rate”: this scalar parameter is a negative multiple of the gradient, and sets the “size” of the step; the higher the learning rate, the higher the variations to the current set of weights and biases at each step. Very high values of the learning rate determine numerical instability, while too small values slow down the convergence process, incrementing the computational cost.

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In batch mode, all the inputs are applied to the network before the weights are updated.

Since the batch training is the most commonly used, this section describes only the batch mode of training; more information about the two training styles can be found in the previous Paragraph 2.3.4.

In batch mode, the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases. For a discussion of batch training with the backpropagation algorithm, see page 12-7 of Hagan et al. (1996).

Within the batch steepest descent training function, the weights and biases are updated in the direction of the negative gradient of the performance function. These settings can be set up by defining the training function.

The "Gradient Descent with Momentum" represents a variation of the gradient descent. This algorithm allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low pass filter, momentum allows the network to ignore small features in the error surface. Without momentum, a network can be stuck in a shallow local minimum. With momentum, a network can slide through such a minimum. See page 12-9 of Hagan et al. (1996) for a discussion of momentum.

2.3.5.2 High performance algorithms based on Feedforward-Backpropagation: heuristic techniques

The previous section 2.3.5.1 presented two backpropagation training algorithms: gradient descent, and gradient descent with momentum. These two methods are often too slow for practical problems. This section discusses several high-performance algorithms that can converge from ten to one hundred times faster than the algorithms discussed previously. All the algorithms in this section operate in batch mode.

These faster algorithms fall into two categories. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. One heuristic modification is the momentum technique, which has been presented in the previous Paragraph 2.3.5.1. This Paragraph discusses two more techniques that are heuristic:

1. Variable Learning Rate Backpropagation

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

It is possible to improve the performance of the steepest descent algorithm if the learning rate is allowed to change during the training process. An adaptive learning rate attempts to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

An adaptive learning rate requires some changes in the training procedure. First, the initial network output and error are calculated. At each epoch, new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated.

As with momentum, if the new error exceeds the old error by more than a predefined ratio, (typically 1.04), the new weights and biases are discarded. In addition, the learning rate is decreased (typically by multiplying by 0.7). Otherwise, the new weights, etc., are kept. If the new error is less than the old error, the learning rate is increased (typically by multiplying by 1.05).

This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it is decreased until stable learning resumes.

It is also possible to combine the adaptive learning rate with the momentum training.

2. Resilient Backpropagation

Multilayer networks typically use sigmoid transfer functions in the hidden layers (see Paragraph 2.3.3). These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero, as the input gets large. This causes a problem when the steepest descent is used to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

The purpose of the Resilient Backpropagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor whenever the derivative with respect to that weight changes sign from the previous iteration. If the derivative is zero, the update value remains the same. Whenever the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for several iterations, the magnitude of the weight change increases. A complete description of the Resilient Backpropagation algorithm is given in Riedmiller and Braun (1993).

The Resilient Backpropagation is generally much faster than the standard steepest descent algorithm. It also has the nice property that it requires only a modest increase in memory requirements. It is not required to store the update values for each weight and bias, which is equivalent to storage of the gradient.

The second category of fast algorithms uses standard numerical optimization techniques. (see Chapter 9 of Hagan et al. (1996) for a review of basic numerical optimization.) The next Paragraph 1.3.1.3 presents three types of numerical optimization techniques for neural network training.

2.3.5.3 High performance algorithms based on Feedforward-Backpropagation: standard numerical optimization techniques

How anticipated in the following Paragraph 2.3.5.2, this section presents three types of numerical optimization techniques for neural network training.

1. Conjugate Gradient Algorithms

The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient), the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the Conjugate Gradient Algorithms, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

In most of the training algorithms discussed up to this point, a learning rate is used to determine the length of the weight update (step size). In most of the conjugate gradient algorithms, the step size is adjusted at each iteration. A search is made along the conjugate gradient direction to determine the step size that minimizes the performance function along that line. See page 12-14 of Hagan et al. (1996) for a discussion of conjugate gradient algorithms and their application to neural networks.

2. Quasi-Newton Algorithms

Newton's method is an alternative to the conjugate gradient methods for fast optimization. The basic step of Newton's method is:

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\boldsymbol{A}_k^{-1}\boldsymbol{g}_k,$$

Eq. 2.6

where $\boldsymbol{A}_k^{-1}$ is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases. Newton's method often converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to compute the Hessian matrix for feedforward neural networks.

There is a class of algorithms that is based on Newton's method, but which does not require calculation of second derivatives. These are called quasi-Newton (or secant) methods. They update an approximate Hessian matrix at each iteration of the algorithm. The update is computed as a function of the gradient.

The quasi-Newton method that has been most successful in published studies is the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update. The BFGS algorithm is described in Dennis and Schnabel (1983). This algorithm requires more computation in each iteration and more storage than the conjugate gradient methods, although it generally converges in fewer iterations. The approximate Hessian must be stored, and its dimension is *n* x *n*, where n is equal to the number of weights and biases in the network.

For very large networks, it might be better to use Resilient Backpropagation or one of the Conjugate Gradient algorithms. For smaller networks, however, this algorithm can be an efficient training function.

Because the BFGS algorithm requires more storage and computation in each iteration than the conjugate gradient algorithms, there is need for a secant approximation with smaller storage and computation requirements. The One Step Secant (OSS) method is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm does not store the complete Hessian matrix; it assumes that at each iteration, the previous Hessian was the identity matrix. This has the additional advantage that the new search direction can be calculated without computing a matrix inverse.

The one-step secant method is described in Battiti (1992). This algorithm requires less storage and computation per epoch than the BFGS algorithm. It requires slightly more storage and computation per epoch than the conjugate gradient algorithms. It can be considered a compromise between full quasi-Newton algorithms and conjugate gradient algorithms.

3. <u>Levenberg-Marquardt algorithm</u>

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as $\boldsymbol{H}=\boldsymbol{J}^T\boldsymbol{J}$ and the gradient can be computed as $\boldsymbol{g}=\boldsymbol{J}^T\boldsymbol{e}$, where **J** is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors.

The Jacobian matrix can be computed through a standard backpropagation technique (see Hagan and Menhaj, 1994) that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1}=x_k-\left[J^TJ+\mu I\right]^{-1}J^Te$$

Eq. 2.7

When the scalar μ of Eq. 2.7 is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm

The original description of the Levenberg-Marquardt algorithm is given in Marquardt (1963). The application of Levenberg-Marquardt to neural network training is described in Hagan and Menhaj (1994) and starting on page 12-19 of Hagan et al. (1996). This algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights).

The main drawback of the Levenberg-Marquardt algorithm is that it requires the storage of some matrices that can be quite large for certain problems. Even if a memory reduction technique is used, the Levenberg-Marquardt algorithm will always compute the approximate Hessian matrix, which has dimensions *n* x *n*, where *n* is the number of weights and biases in the network. If the network is very large, it might run out of memory. In these cases, the use of one of the conjugate gradient algorithms is advisable.

2.3.6 Input and output Processing Functions

This Paragraph aims to sketch some routines to be implemented in order to speed-up or improve the ANN performance and the required computational time.

ANN input processing functions transform inputs into a “better form”, i.e. easier and more efficient, for the network use. Processing functions associated to ANN outputs transform targets into a better form for network training, and reverse transformed outputs back to the characteristics of the original target data. It is then possible to provide the network with both “pre-processing” (for inputs) and “post-processing” (for outputs) functions.

Among pre-processing, one of the most common and useful function (the function “*mapminmax*” in software *Matlab*) transforms input data so that all values fall into the interval [-1; 1]. This can speed up learning for many networks. Another function (“*removeconstantrows*”) removes the values for input elements that always have the same value because these input elements are not providing any useful information to the network. A third common pre-processing function (“*fixunknowns*”) recodes unknown data (represented in the user's data with “Not a Number” – or NaN – values) into a numerical form for the network, preserving information about which values are known and which are unknown.

Similarly, output post-processing functions are used to transform user-provided target vectors for network use. Then, network outputs are reverse-processed using the same functions to produce output data with the same characteristics as the original user-provided targets. Both “*mapminmax*” and “*removeconstantrows*” are often associated with network outputs.

2.4 Conclusions

This Chapter has explained the conceptual basis of the neural networks modelling, in order to provide the reader with an idea of what Artificial Neural Networks (ANNs) are, before illustrating the literature examples of such models and presenting the new ANN developed within this work.

The main structures and working principles characterizing an ANN have been presented and described. More attention has been paid on those features that will have been employed for the realization of the new ANN presented in this thesis (see Chapter 4).

An ANN is essentially a numerical tool that is very useful for solving classification and regression problems. It has been stressed that such models actually belong to the category of “black-boxes”. The intrinsic idea of the method is to imitate the behavior of an animal brain. Input information arrives to so-called neurons after processing this information between all the interconnected neurons, and a final result is given as output.

A basic aspect of an ANN is the learning process. Learning is the process that calibrates and determines the value of weights and biases, the internal parameters of the ANN that allow the model to properly reproduce the input-output relationships. Starting with small random initialization values of such parameters (weights and biases), the network processes the inputs. The resulting output of a network generally deviates from the desired output. The goal of the learning process is to adapt the weights and biases in such a way, that the difference between the desired and calculated output becomes smaller. By iteratively repeating the learning, the proper values of weights and biases could be approached. The overall iterative process is called training.

There are different types of training methods available. The choice of an appropriate training method essentially depends on the problem that is faced. This Chapter has presented an overview of the most commonly used training algorithms.

An associated problem appears during training. This is known as the “generalization problem”, or “over-training” risk. An ANN is said to be “over-trained” when it fits very well the data used to train the network, but the predicting of new data results in a large error. To solve the problem of over-training, a generalization method can be applied to the ANNs. The most important generalization techniques have been discussed, anticipating the importance of the bootstrap resampling.

3. STATE OF THE ART: ARTIFICIAL NEURAL NETWORKS MODELLING WAVE-STRUCTURE INTERACTION

3.1 Introduction

Properly trained on an adequate database, the ANNs demonstrated to be powerful tools able to satisfactory fit several practical physical and engineering applications. A wide analysis and description of this approach can be found in Cherkassky et al. (2006), who introduced a specific dedicated section in the Neural Networks Journals.

Within Ocean and Atmospheric field, the adoption of ANNs has rapidly increased during the last years. One of the first applications of neural networks is represented by the so-called "hybrid models", i.e. numerical models partially re-calibrated after training with ANNs. In some cases, the ANNs were developed in order to improve the overall performance of an existing numerical model by replacing only part of the complete code. An example of this particular approach – evolved in the so-called "hybrid-modelling" – is the "*Neural Network Interaction Approximation*" (NNIA, Krasnopolsky and Chevalier, 2003; Tolman et al., 2005), an ANN-based algorithm designed to substitute the "*Discrete Interaction Approximation*" (DIA) of non-linear wave interactions within spectral models.

ANNs were firstly applied in Coastal Engineering for the prediction of time series of wave parameters in a specific place of interest (usually, next to the shoreline) starting from elsewhere detected time series (usually offshore), for example by satellite measurements (Kalra et al., 2005), by buoys wave signals (Tsai et al., 2002; Makarynskyy et al., 2005-a), by results obtained from large scale models or wind data (Browne et al. 2006, Deo et al. 2001; Rao and Mandal, 2005).

This technique was also employed to predict future values of wave parameters on the basis of previously measured values of the same parameters (see, for example, Deo and Naidu, 1999; Makarynskyy et al., 2004-a; Makarynskyy et al., 2004-b; Londhe and Panchang, 2006), to interpolate lacking values (Makarynskyy et al., 2005-b), to study some interdependences between wave parameters (Agrawal and Deo, 2004) and to increase the precision of results achieved from wave simulation numerical models (Makarynskyy et al. 2005-b; Zhang et al., 2006).

The earliest examples of the application of the neural network modelling within the field of Coastal and Ocean Engineering see the introduction of this approach as a substitution for more traditional techniques of numerical modelling. Within the main important applications, the following authors could be mentioned:

- Mase et al. (1995) analyzed the applicability of ANNs for predicting the stability of
- rubble mound breakwaters;
- Van Gent and Van den Boogard (1998) developed an ANN for the prediction pf wave forces on vertical structures;
- Medina (1999), Medina et al. (2002, 2003) focused on the prediction of the wave run-up and wave overtopping;
- Panizzo et al. (2003), Pozueta et al. (2004), Van Oosten and Peixó Marco (2005) investigated the wave transmission.

Concerning the specific field of the wave-structure interactions, one of the most recent and successful ANNs is the method available from CLASH (2004) and EurOtop (2007) for wave overtopping, which was specifically developed to predict the overtopping discharge for a wide range of coastal structures, including complex geometries (Van Gent et al., 2007).

After and during CLASH (2004) other ANNs were developed. Specific ANNs were developed for the estimation of the wave overtopping discharge (Verhaeghe et al., 2008), of the wave transmission coefficient (Van Oosten and Peixó Marco, 2005; Panizzo and Briganti, 2007) and of the wave reflection coefficient (Zanuttigh et al., 2013), resulting a valid alternative to more traditional techniques. Each of these ANNs actually proved to be able to overcome some of the limits imposed by the traditional empirical formulae.

The design of coastal and harbour structures requires indeed a systematic analysis of all the processes of wave-structure interaction, which takes into account the combined effects of wave overtopping, wave transmission and wave reflection. The development and/or use of an Artificial Neural Network (ANN) is therefore particularly recommended in case of complicated structure geometries and variable wave conditions.

This chapter presents the most important and recent applications of ANNs for the modelling of wave-structure interaction processes. Starting with paragraph 3.2, a brief *excursus* on the first applications of ANNs within Coastal Engineering is presented. Then, focusing on the more-specific case of wave-structure interactions, paragraphs from 3.3 to 3.5 respectively illustrate the state of the art relative to the prediction of the wave overtopping discharge, the wave transmission coefficient and the wave reflection coefficient.

3.2 First ANNs for Coastal and Ocean Engineering

This section aims to present some of the most relevant ANN applications in Coastal Engineering, before focusing on the recent models specifically developed for the prediction of the main wave-structure interaction parameters (i.e. the wave overtopping discharge, q, the wave transmission and the wave reflection coefficients K_t and K_r).

Besides the ones reported in the following sub-sections (from 3.2.1 to 3.2.5), many other ANN applications in Coastal Engineering can be referred to. More examples of the earliest ANN research performed in various fields of Coastal Engineering are:

- tidal level forecasting (Tsai et al., 1999);
- prediction of the occurrence of impact wave force (Mase et al., 1999);
- analysis of wave directional spreading (Deo et al., 2002);
- prediction of storm-built beach profile parameters (Tsai et al., 2000);
- prediction of scour depths at culvert outlets (Liriano et al., 2001),
- prediction of wind induced water levels (Westra et al., 2002);
- prediction of sedimentation in the Maasmond (Bierens, 2002);
- prediction of the breaker depth and breaking height of breaking waves (Deo et al., 2003);

These references are definitely not meant to give a complete overall view of performed ANN research in Coastal Engineering during the last years. The aim is rather to give an idea of the various subjects for which ANNs may be used.

The studies mentioned in the following sub-sections will show that ANNs have applications in very different research domains. Complex relationships in various research fields may be modelled with ANNs, on condition that enough measurements are available to calibrate the neural model. ANNs are often used if the derivation of reliable empirical relations based on measurements is difficult due to the complex relationships. It was proved that the prediction capacity of an ANN is largely dependent on the quality of the data on which it was trained (e.g. Van Gent and Van den Boogaard, 1998). As mentioned in the studies above, existing empirical design formulae are often used to give an idea of the prediction capacity of the developed network.

Please note that this introductive paragraph is partially based on Verhaeghe (2005) and Formentin et al. (2012).

3.2.1 Spectral modelling combined with neural network methods for the wind-wave variability

Herman et al. (2009) conduced an analysis of the wind-wave variability in the tidal basins of the German Wadden Sea. Because of the high computational costs associated to the traditional models, the necessity to develop an alternative technique, such as the neural networks, rose for the reproduction of the hydrodynamic processes in the coastal areas.

Herman et al. employed the high-resolution results of the simulations of the wind generated waves as starting point for the realization of an ANN-based model which allowed more speedy and more accurate estimations of the spatial distribution of the significant wave heights (H_s), periods ($T_{m-1,0}$) and directions (θ_m).

The wave propagation and transformation in the study area were modelled with the state-of-the-art third-generation spectral wave model SWAN. The principal component analysis of the SWAN results was then used to reveal the dominating spatial patterns in the data and to reduce their dimensionality, thus enabling an efficient and relatively straightforward ANN modelling of mean wave parameters in the whole study area-

The architecture of the ANN was an MLP back-propagation feed-forward network and consisted of one hidden layer including 90 hidden neurons. A hyperbolic tangent function was used as transformation function for the hidden layer, and a linear one for the output layer. The adopted training algorithm belonged to the class of gradient descent algorithms.

The 14 input parameters comprehended the significant wave height period and direction, the water level, the two components of the wind velocity and the time-averages of those components. The output layer contained alternatively 5 or 10 neurons according to the time-series to be predicted: 5 neurons in case of significant wave height and wave period time-series and 10 neurons for the significant wave direction.

The results of the application provided a satisfactory result. The combination of spectral and neural network modelling for wind-generated waves proved to represent a valid alternative to the classic approach. An important outcome of such methodology is the possibility to be applied in similar regions. Some tests carried out on other sites on the German coast have indeed demonstrated that the same ANN architecture and the same input parameters could be employed to simulate these areas.

Finally, the ANN could have provided a better performance if longer time-series were available for the training and the validating processes. This aspect stresses the importance of the wideness and homogeneity of the training database, a point that will be further discussed in the present work.

3.2.2 Analysis of the stability of rubble mound breakwaters by neural network modeling

Mase et al. (1995) analyzed the applicability of ANNs for predicting the stability of rubble mound breakwaters.

An MLP (see paragraph 2.3) with one hidden layer was calibrated with a dataset consisting of 100 data, originating from van der Meer (1988) experimental data. A modified momentum method was applied for the learning process. Seven input parameters concerning the stability of rock slopes were proposed.

For a first ANN, the output parameter was the damage level $S = A/D^2_{n,50}$ (where A is the eroded area of the breakwater cross-section and $D_{n,50}$ is the nominal diameter of the stone) was used as output parameter, whereas for a second ANN the output parameter was the stability number $N_s = H_s/(\Delta \cdot D_{n,50})$ (where H_s is the significant wave height and Δ is the relative density of the rocks). The predicted damage levels by the ANN agreed satisfactorily well with the measured damage levels of data which were not used for the training process, i.e. a part of the van der Meer (1988) dataset and the data of Smith et al. (1992). The agreement between the stability numbers predicted by the ANN and the measured ones was also found to be good, but not better than the stability formula of van der Meer (1988) itself.

3.2.3 Prediction of wave forces on vertical structures

Van Gent and Van den Boogaard (1998) used the neural network modelling to predict horizontal forces on vertical structures. The horizontal force exceeded by 99.6% of the waves, $F_{h-99.6\%}$, was considered. A dataset composed of 612 data resulting from model tests performed in 5 different laboratories was used. An MLP with 1 hidden layer was calibrated with the standard backpropagation method (see paragraph 2.3.5.1). The network consisted of 9 input parameters corresponding to the main factors determining the total horizontal wave force on vertical structures.

The performance of the neural model was compared to the formula of Goda (1985), which showed a better performance of the neural model for the considered data.

In addition, Van Gent and Van den Boogaard (1998) developed a method to describe the reliability of the ANN prediction. This methodology enlightened an important issue, i.e. that inconsistencies in the database may largely influence the prediction method.

3.2.4 Preliminary studies for the prediction of wave run-up and wave overtopping trough ANNs

Medina (1999) and Medina et al. (2002) studied the ANN modelling of run-up and overtopping.

Medina (1999) used an Evolutionary Strategy (ES) to optimize the parameters and the topology of two ANNs. Both the ANNS consisted in MLPs with one hidden layer: one was developed to predict the wave run-up of regular waves at a conventional rubble mound breakwater; the second one to predict the wave run-up at a dissipating basin breakwater. The experimental data described by Medina (1998) and González and Medina (1999) were used for the calibration of the models, i.e. 826 tests with regular waves on a conventional breakwater and 1250 tests with regular waves on a dissipating basin breakwater. A part of the tests was performed with artificial wind generation in laboratory.

The predictions obtained with the ANN for the conventional breakwater were found to be reasonably accurate. The predictions obtained with the ANN for the dissipating basin breakwater were instead poorer. No further comparison with existing formulae was performed.

In Medina et al. (2002) the same ES was used to calculate a chain of two pruned neural models able to detect significant overtopping (i.e. $q > 10^{-4.5}$ l/m/s) and to estimate overtopping discharges at a rubble mound breakwater. Two MLPs with each one hidden layer and four input parameters were proposed. A number of 113 tests with irregular waves, of which a part was performed with wind generation, (see Medina et al., 2001) served for the calibration of the models. The models proved to be efficient for the considered data. Here as well no comparison with existing empirical formulae was performed.

It should be mentioned that, comparable to the final approach of the overtopping prediction method proposed by Verhaeghe et al. (2008), Medina et al. (2002) firstly presented two subsequent neural models to predict overtopping. The first ANN 'classified' the overtopping and the second one 'quantified' significant overtopping.

3.2.5 Preliminary studies for the prediction of wave transmission trough ANNs

Panizzo et al. (2003) calibrated a neural model with experimental data on wave transmission over rubble mound low-crested structures. The reference datasets used were the data gathered within the EC project DELOS, from which 5 subsets of data were defined, for a total number of approximately 2140 data. An MLP with one hidden layer was calibrated using the Levenberg-Marquard algorithm. Six dimensionless parameters, related to hydraulic as well as structural parameters, were proposed as input. The introduction of non-dimensional parameters represents a novelty and an important achievement, which will be adopted also for the new advanced ANN, presented in this work.

A good performance of the developed neural model for the transmission coefficient K_t was found. Comparison with existing empirical formulae showed that the neural model results were more accurate.

Van Oosten e Peixó Marco (2005) provided an ANN model which proved to be capable of handling both smooth and mound structures, although their behavior against the wave transmission is completely different. The advantaged of handling both structures in one prediction model is that also structures in the transition between smooth and mound structure can be handled. Examples for instance are an impermeable mound structure or a reduced smooth structure. For this reason, this study represented one of the first attempts of extending the ranges of validity of an ANN, in order to develop a predicting method which can be adopted for a wider field of applications. Another important distinctive characteristic of this ANN was the adoption of the strategy of "committee of networks" (see paragraphs 2.2.2 and 2.2.3) originally proposed by Verhaeghe (2005) (see paragraph 3.3.1).

The ANN by Van Oosten e Peixó Marco (2005) was characterized by twelve dimensionless input parameters describing the structure features and the wave attack conditions. The architecture consisted of a MLP with one hidden layer of 17 hidden neurons; the training method was based on the Bayesian regularization. The performance of the model was compared to the EC project DELOS formulae and provided satisfactory results.

3.3 Prediction of the wave overtopping discharge: the CLASH ANN

For design, re-arrangement, risk assessment and safety analysis of coastal and harbour structures, a reliable and accurate prediction of the wave overtopping discharge (q) is required.

The physical relationships between q and the parameters characterizing the wave overtopping process are highly non-linear and not easily representable by analytical formulations. The number of the involved parameters is elevated, since the proper description of the overtopping discharge strictly depends not only on the wave attack conditions (representable through the wave height, period, direction) but also on the structure geometry itself, which may be very complicated indeed (just consider the potential presence of toe protections, one or more berms, crown walls, etc.). Consequently, the analytical formulae for the evaluation of q (e.g. EurOtop 2007, van der Meer et al., 2013) become rather complex and strictly dependent on empirically defined coefficients, whose values are derived from specific datasets and therefore vary from one case to another one. Furthermore, the experimental measurements on which the formulae are fitted and validated might be affected by random or systematic uncertainty, not always properly assessed.

Then, the actual reliability of most of the empirical formulae is not defined and several differences so far exist among the predicted values of q derived from different formulae.

The European Project CLASH ("*Crest Level Assessment of coastal Structures, by full scale monitoring, neural network prediction and Hazard analysis on permissible wave overtopping",* see CLASH, 2004) was born within this background, setting its main goals to:

- analyze the scale-effects for the wave overtopping;
- realize an ANN-based predicting method to evaluate the wave overtopping discharge and assessing the uncertainty of the prediction.

The CLASH project aimed to achieve a predicting model that could fit as many different typology of structures as possible, both in prototype and laboratory scale, in order to include also non-standardly shaped constructions. For this purpose, a great number of parameters describing the structure geometry were required, and therefore a great number of empirical tests needed to be performed and collected.

Within CLASH, more than 10'000 measurements of overtopping discharge were gathered from experimental laboratory and prototype tests carried out in several different facilities. The total amount of data was collected within an unique wide database, originally called "*The new overtopping database for coastal strucutres*" (general presentation in Steendam et al, 2004; Verhaeghe et al., 2008; armour unit data in Bruce et al., 2006; van der Meer et al., 2008) and subsequently "*The overtopping database*".

Since the database was thought to be employed also for other applications, the greatest number possible of data for each test was gathered. It was not just registered the essential information (i.e. the wave and structural parameters and, of course, q), but also detailed records on the implemented techniques of measurement of waves and discharges and on the adopted analysis methodologies.

For each test, three groups of parameters were defined: 3 general parameters, 11 hydraulic parameters and 17 structural parameters. The general parameters consist of an unique name for each test and the reliability and complexity factors, *RF* and *CF*. The hydraulic parameters describe the wave characteristics and the measured overtopping, whereas the structural parameters describe the tested structure (as depicted in Fig. 3.1). In particular, *CF* and *RF* relate to the degree of approximation pursued in the description of the cross-section layout of a tested structure. *CF* accounts for the geometry complexity, while *RF* the reliability itself associated to the measurement. *CF* and *RF* values may vary between 1 and 4, being 1 the simplest (*CF*) or most reliable (*RF*) condition and 4 the “worst” case.

A more detailed information about the database can be found in Paragraph 4.2, where the CLASH database scheme and approach is fully described.

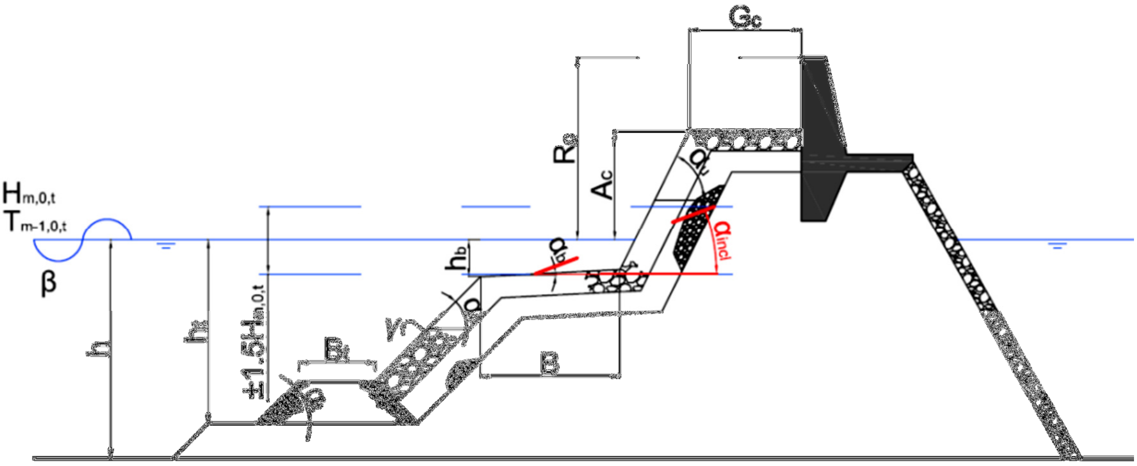


Figure 3.1 – Structural and hydraulic parameters involved in the CLASH database.

Figure 3.1 shows the most general possible layout of a structure cross-section, employing the same symbolism also adopted within CLASH database to describe the hydraulic and geometrical parameters. The meaning of the parameters shown in figure is listed in the following:

- h: water depth in front of the structure, [m];
- $H_{m,0,t}$ and $T_{m-1,0,t}$: wave main height [m] and spectral period [s], at the toe;
- h_t: toe submergence, $[m]$; in absence of toe, it is equal to h, the sea depth in front of the structure;
- h_b: berm submergence, $[m]$; if the structure has neither berms nor toes, it is equal to h;
- $cot\alpha_d$: slope of the lower part of the structure, below the berm if it is present, [-];
- $cot\alpha_u$: slope of the upper part of the structure, above the berm if it is present, [-];
- $cot\alpha_b$: slope of the berm if it is present, [-];
- $cot\alpha_{incl}$: mean slope within the run-up and run-down zone, including the berm; the mean angle α_{incl} is defined between $\pm H_{m,0,t}$, [-]; in case of straight slopes, $\alpha_{incl} = \alpha_d$;
- γ_f: armour roughness coefficient [-];
- R_c: breakwater freeboard, eventually comprehensive of the crown sea wall, [m];
- A_c: breakwater freeboard non-comprehensive of the eventual crown sea wall, [m];
- B: berm width, $[m]$;
- G_c: breakwater crest width, $[m]$;
- m: foreshore slope, defined as the cotangent of the angle of the slope itself, [-];
- β: angle of deviation from the perpendicular wave attack direction, [°];

The database of CLASH had a much larger number of measurements than any other previous case and was suitable for an ANN model training. Van Gent, et al. (2007), in agreement with CLASH project purpose, first realized an ANN model for the prediction of the specific wave overtopping discharge (q, m^3/(sm)) based on the CLASH database. However, the total amount of about 10’000 data was reduced to approximately 8’300 data, since all the data presenting either a *RF* or a *CF* value equal to 4 were considered “not reliable” and discarded. Furthermore, also the tests presenting a value of q identically 0 (cases of “zero-overtopping”) or lower than a threshold value of q=10^{-6} m^3/(sm) were discarded as well. The threshold was set in order to distinguish among "significant" and "non-significant" wave overtopping for design purpose.

Essentially, the ANN proposed by Van Gent, et al. (2007) was focused on the prediction of the "significant" wave overtopping discharge, rejecting the possibility to deal also with zero-overtopping cases. Verhaeghe et al. (2008) later suggested a "correction" to this approach, through the coupling of a classifier-quantifier ANN (see Paragraph 3.3.1).

As already stated, objective of the CLASH project was that the ANN should work both with model and prototype scale tests. Therefore, the need of scaling all the input parameters and the output parameter q occurred. In order to have, for each test, a wave attack condition characterized by unitary wave height (i.e. $H_{m,0,t}$ = 1m), the scale factor was settled to be the $H_{m,0,t}$ and all the parameters were consequently scaled, according to the Froude similarity law.

The ANN was characterized by only a "*Multi-Layer Perceptron feed-forward*" architecture (see paragraph 2.3) with only one hidden layer, trained with the "*standard error back-propagation rule*" (see Paragraph 2.3.5). Figure 3.2 reports the architecture of the ANN, showing the 15 input elements (the terminology adopted within this figure refers to the input vector as "input layer", which is an improper denomination, see Paragraph. Xx), the 20 hidden neurons and the output neuron, q.

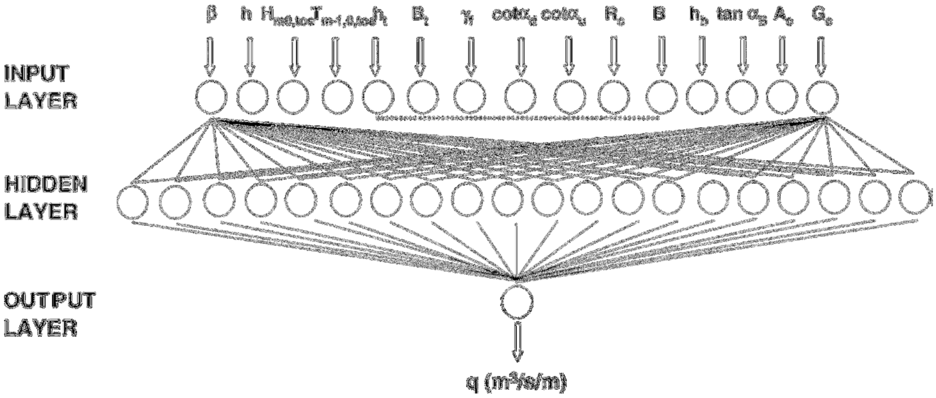


Figure 3.2 – Architecture of the CLASH ANN for the prediction of the wave overtopping discharge.

The input vector was composed by the following 15 input elements:

$$[H_{m0,toe}, T_{m-1,0,toe}, \beta, h, h_t, B_t, \gamma_f, \cot\alpha_d, \cot\alpha_u, B, h_b, \tan\alpha_b, R_c, A_c, G_c] \,. \tag{Eq. 3.1}$$

The optimal number of *hidden neurons* was definitely set to 20 after a calibration process (see paragraph 2.2.2): the ANN was trained and tested several times, each time varying the number of hidden neurons and computing the corresponding performance function (both for training and testing set). As it can be detected from Figure 3.3, typically the performance function exponentially decreases for a limited number of hidden neurons, and then tends to assume an asymptotic profile when the "over-training" occurs (see paragraph 2.2.2). This phenomenon depends on the ANN architecture's complexity. When a too few hidden neurons are selected, the model cannot properly reproduce the experimental input-output pattern due to a too poor number of connections among the neurons; when a sufficiently-enough (in this case, evidently 20) number is chosen, and no further improvements can be reached, the model do not benefit of any increased complexity of its architecture.

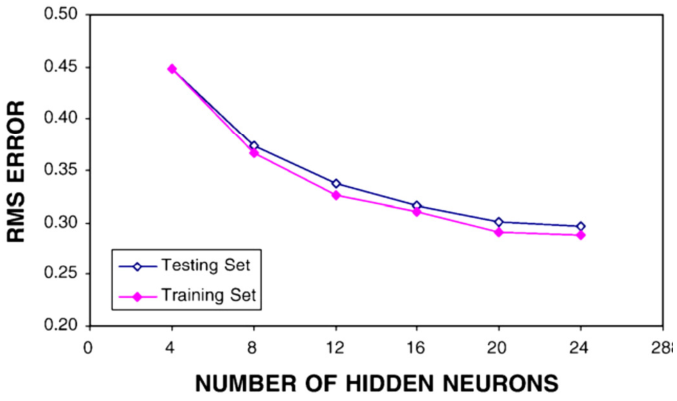


Figure 3.3 – CLASH ANN performance function (ordinate) at the increasing number of hidden neurons (abscissa). Training and testing sets are distinguished.

The ANN was trained by separating the database into three datasets (training, testing and validating, employing the "early-stopping" technique for the generalization of the model, see Paragraph 2.2.2). The definition of the weights and the biases values during the learning phase was based on the minimization of a performance function, the root mean squared error (*RMSE*) of the differences among the target values of q and the corresponding predicted outputs. In details, the *RMSE* was computed following Eq. 3.2,

$$RMSE=\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left((\log q'_{obs})_n-(\log q'_{ANN})_n\right)^2}\ ,$$

Eq. 3.2

where N is the number of tests used for training the ANN. From eq. 3.2, it can be observed that the ANN was actually trained on the logarithm of the overtopping discharge, i.e. the target values (the observed discharges q_{obs}) was processed to the network after a logarithm transformation and consequently the ANN predicted logarithmic values (q_{ANN}). The superscript signs (log q'_{obs} *or log* q'_{ANN} *stand to refer to the* Froude-scaled values of q.

The resulting *RMSE* associated to the CLASH ANN predictions was used by Van Gent, et al. (2007) as a quantitative index of performance. This *RMSE* value is reported in eq. 3.3-3 (please, note that the value relates to the logarithmic transformations of q).

$$RMSE=0.29$$

Eq. 3.3

The value of *RMSE* is considered satisfactory small (see Van Gent et al., 2007), also in comparison with existing traditional formulae and approaches. To better characterize the quality of the predictions, a plot comparing the predictions with the corresponding measurements of q was realized (here reported in Fig. 3.4).

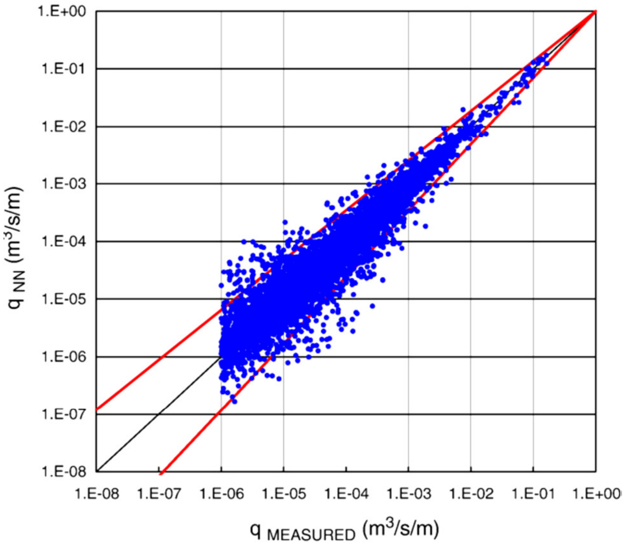


Figure 3.4 – ANN predicted values of overtopping discharge (q_{NN}, ordinate) as a function of the corresponding measured values ($q_{MEASURED}$, abscissa). The bisector line represents the ideal condition of $q_{NN} = q_{MEASURED}$, while the red lines indicate the 90% confidence interval.

From this plot, the predictions result accurate and not excessively biased, especially for high values of the measured *q*, while for low values of *q* a tendency to overestimation is observed. The red lines in Figure 3.4 represent the 90% confidence interval bands, i.e. the bands within the 90% of the predicted values fall. These bands, computed as shown in Eq. 3.4, were added to the plot in order to illustrate the effective spreading of the predictions around the expected values.

$$\log q_{NN\text{-}90\%} = \begin{cases} 0{,}86 \cdot \log q_{obs} \\ 1{,}15 \cdot \log q_{obs} \end{cases}$$

Eq. 3.4

The reliability of the performance was furtherly assessed by Van Gent et al. (2007) by means of the so-called “resampling techniques”, in particular the *bootstrap* resampling (see Paragraph 2.2.3). The adoption of the bootstrapping allowed the authors to associate to the ANN predictions a distribution of the error: in other words, each predicted *q* corresponds to a quantile of a distribution of values obtained by several (L-times) resampling the database and several re-training and re-running the ANN. Therefore, the ANN supposed user obtains an “average” prediction and the associate standard deviation or, according to his needing, a specific quantile (e.g., the quantiles $q_{2,5\%}$, $q_{5\%}$, $q_{25\%}$, $q_{50\%}$,$q_{75\%}$,$q_{95\%}$, $q_{97,5\%}$).

It is important to remark that in the case of CLASH, the bootstrap resampling was introduced only as a methodology to assess the performance and not as a technique to improve the generalization of the ANN and substitute the “early-stopping” (see Paragraph 2.2.2). Anyway, the uncertainty assessment is an essential aspect for an ANN model, standing its nature of “*black-box*” working on randomly selected experimental input-output pattern and the consequent dependence of the predictions on the random training dataset.

A key issue to be pointed out is the underlying supposition of reliability of the database: the confidence intervals and the quantiles are indeed derived based on the observed values of *q*. Whether random or systematic errors occurred, the model would result affected without possibility of acknowledge.

Generally, the predictions of the wave overtopping discharge derived by Van Gent et al. (2007) ANN, within the context of the CLASH project and using the CLASH database for training, result accurate. Further applications of the ANN demonstrate that the model is also able to evaluate the influence of single input parameters (such as the angle of the wave attack direction *β*, the relative crest-freeboard $R_c/H_{m,0,t}$ or the relative crest width $G_c/H_{m,0,t}$) for a wide range of structures.

A paramount conclusion obtained with the CLASH project is that the quality and the reliability of the predictions is strictly dependent on the reliability of the training database itself.

3.3.1 A two-phase classifier-quantifier ANN for the wave overtopping discharge

The adoption of the logarithmic transformation of the target values of *q* by Van Gent et al. (2007) was due to the effective impossibility for an ANN to work with the observe natural values of overtopping wave discharge, *q*. Indeed, standing an exponential relationship between the crest-freeboard of a structure and the corresponding *q* (e.g., TAW, 2002; EurOtop, 2007), the natural values of *q* can vary of several orders of magnitude (within CLASH database, $q \approx 10^{-9} \div 10^{-1}$ m^3/(sm)). An ANN simply trained on such target would properly deal only with “large” values ($q \approx 10^{-1} \div 10^{-2}$ m^3/(sm)), since, during the training phase, the differences between target and predicted *q* are minimized, and consequently the ANN cannot distinguish among “large” or “small” values. The introduction of the pre-processed *q* values through a logarithmic transformation allows the ANN to work with relative errors of the same order of magnitude. Nevertheless, the values of *q* identically equal to zero still represented a problem, since $log\ (q) = -\infty$, and the elimination of zero-overtopping tests was detected to be the cause of the ANN tendency to overestimate the low values of q.

A few alternatives to the “official” CLASH ANN were then investigated, even if never adopted. One of the most interesting studies, started by Verhaeghe (2005) and concluded in Verhaeghe et al. (2008) analyzed the possibility of coupling two ANNs in order to overcome the exclusion of zero or “small” overtopping tests.

Before Veraheghe (2005), the technique of coupling two ANNs in series had been already adopted by Medina (1998) for the evaluation of the wave run-up over specifically shaped breakwaters for the dissipation of the wave energy.

This methodology was supposed to be suitable also for the prediction of the wave overtopping discharge, in order to solve the matter of predicting small and zero *q* values.

The model proposed by Verhaeghe was composed by two ANNs working in series one after the other. A first ANN should have pre-processed the overall database and should have been trained to produce a Boolean distinction between “significant” (i.e. $q>10^{-6}$ m^3/(sm)) and “non-significant” overtopping cases. The adopted threshold value of 10^{-6} m^3/(sm) was directly derived from Van Gent et al. (2007). A second ANN, would have processed only the tests classified as “significant” by the first one and would have definitely quantified the numerical values of *q*.

This two-phase approach (called classifier-quantifier ANN by Verhaeghe et al. (2008)) would have presented the possibility to deal with the complete database and provide the user of a tool actually being able to predict also zero-overtopping conditions. Furthermore, it would have partially resolved the problem of overestimation of low values of *q* present in the work of Van gent et al. (2005). It essentially represented a development and improvement of the single-ANN model proposed by Van Gent et al. (2007).

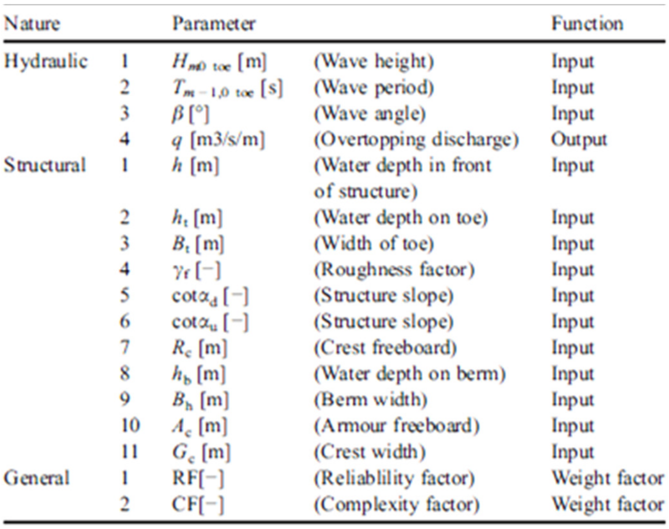
Both the classifier and quantifier ANNs presented a MLP (*Multi-Layer Perceptrons*) architecture, characterized by a single hidden layer and an unique output neuron. A single hidden layer architecture demonstrated to be sufficient for the expected modelling.

The same CLASH database employed by Van Gent et al. (2007) was used. 17 out of the 31 available parameters (see Fig.s 3.5 and 3.6) were selected:

- 13 to constitute the input array (which is the same for both the classifier and the quantifier, since the two ANNs were supposed to work in series);
- 2 as weight factors for the bootstrapping resample (the coefficients *RF* and *CF*);

the last one, the significant wave height $H_{m,0,t}$, as the scaling factor for all the input parameters according to the Froude similarity.

Differently from Van Gent et al. (2007), the parameter B_h, representing the "average" berm slope, replaced *B* and $tan\alpha_B$ (the number of the sloping berms was indeed limited and the elimination of one input parameter reduces the complexity of the ANN architecture).



Nature		Parameter		Function
Hydraulic	1	$H_{m0\ toe}$ [m]	(Wave height)	Input
	2	$T_{m-1,0\ toe}$ [s]	(Wave period)	Input
	3	β [°]	(Wave angle)	Input
	4	q [m3/s/m]	(Overtopping discharge)	Output
Structural	1	h [m]	(Water depth in front of structure)	Input
	2	h_t [m]	(Water depth on toe)	Input
	3	B_t [m]	(Width of toe)	Input
	4	γ_f [−]	(Roughness factor)	Input
	5	cotα_d [−]	(Structure slope)	Input
	6	cotα_u [−]	(Structure slope)	Input
	7	R_c [m]	(Crest freeboard)	Input
	8	h_b [m]	(Water depth on berm)	Input
	9	B_h [m]	(Berm width)	Input
	10	A_c [m]	(Armour freeboard)	Input
	11	G_c [m]	(Crest width)	Input
General	1	RF[−]	(Reliablility factor)	Weight factor
	2	CF[−]	(Complexity factor)	Weight factor

Figure 3.5 – Input parameters composing the classifier and the quantifier ANNs.

A different number of hidden neurons composed the hidden layers: 20 for the classifier ANN, 25 for the quantifier. A larger number of hidden neurons was require for the quantifier hidden layer, due to the more complex output to be predicted. The classifier ANN simply had to perform a logical value as output: +1, in case of "significant" *q* (i.e. $q>10^{-6}$ $m^3/(sm)$), otherwise -1. The quantifier output was instead a numerical value of *q*, the effective evaluated overtopped discharge (scaled according to Froude and logarithmic transformed). The layout of the quantifier ANN is sketched in Fig. 3.6.

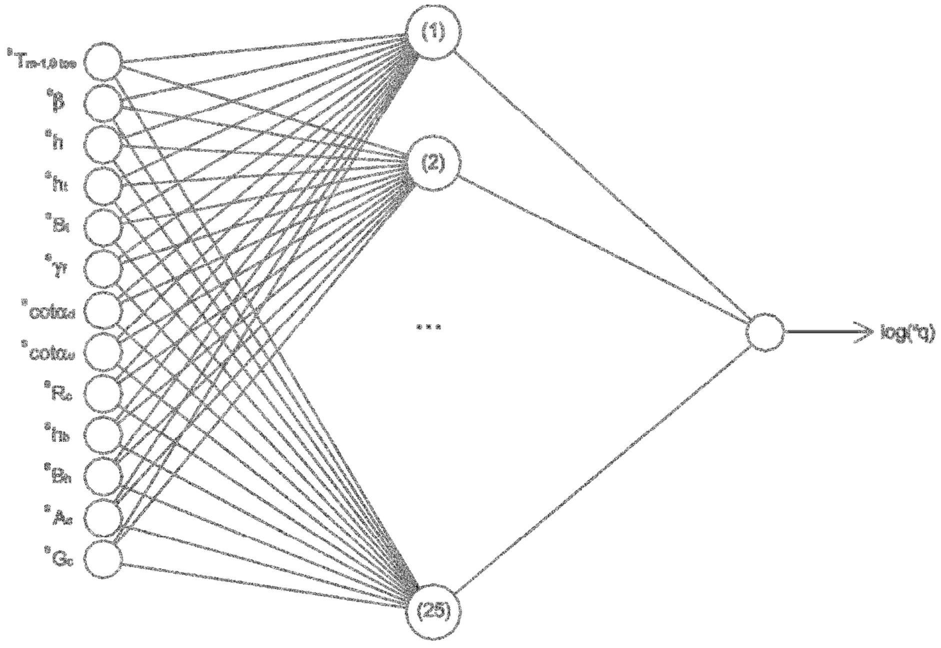


Figure 3.6 - Architecture of the quantifier of the two-phase classifier-quantifier ANN developed by Verhaeghe et al. (2008).

At first (Verhaeghe, 2005), the early-stopping technique (see Paragraph 2.2.2) was adopted for both the ANNs, following Van Gent et al. (2007). The database was partitioned, by assigning a random 85% of the total amount for training and the remaining 15% for the validating and testing sets.

The employed performance function was the root mean squared error (*RMSE*), defined as stated in eq. 3.5:

$$\text{RMSE}=\sqrt{\frac{1}{N}\sum_{n=1}^{N}[(o_{obs})_n-(o_{NtN})_n]^2}\ ,$$

Eq. 3.5

where *N* corresponds to the total number of pairs of measured-predicted values of *q*, o_{obs} is the target value corresponding to the nth out of *N* tests and o_{NN} is the ANN predicted output.

Later, the bootstrap resampling (see Paragraph 2.2.3) was introduced, not only to assess the performance, but also as a generalization methodology, excluding the early-stopping and the partition of the database. The bootstrap allowed achieving an improved performance, if compared to the case of early-stopping (Verhaeghe, 2005).

Figure 3.3.7 shows the outcome of the associated classifier and quantifier ANNs, which corresponds a quantitative index of performance of *RMSE=0.31.*

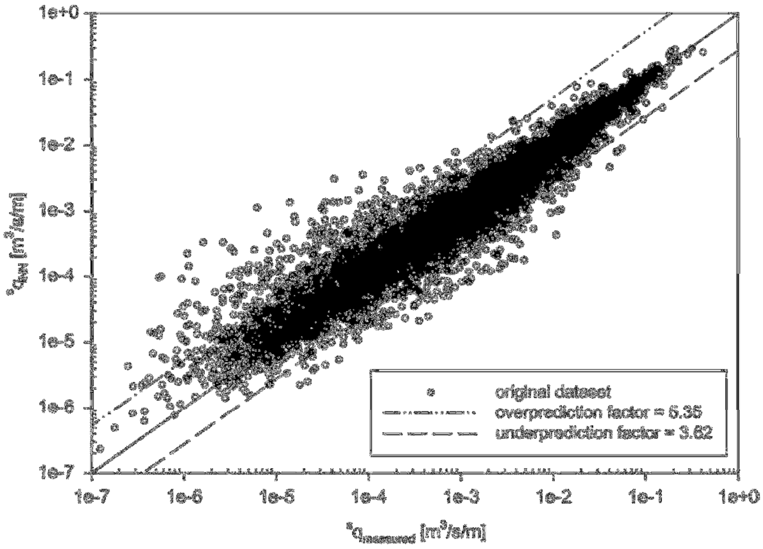


Figure 3.7 – Comparison among the values of overtopping discharge (q_{NN}, ordinate) predicted by the coupled classifier-quantifier ANNs and corresponding measured values ($q_{measured}$, abscissa).

The comparison between the CLASH ANN performance (*RMSE*=0.29, see Paragraph 3.3) and the two-phase approach by Verhaeghe et al. (2008) is not straightforward, since the latter involves also the prediction of small and zero-values of *q*. In other words, the qualitative (Fig. 3.4 and Fig. 3.7) and the numerical (*RMSE*=0.29 against *RMSE*=0.31) comparison are not consistent, since the CLASH ANN processed about 8’000 tests, all relating to cases of q>10^{-6} m^3/(sm), while Verhaeghe et al. (2008) ANN employed the nearly complete CLASH database (the cases of *RF* and *CF* = 4 are still excluded), about 10’000, comprehending each value of *q*.

Then, a significant comparison may be carried out by accounting of the percentage of wrongly classified tests, i.e. the percentage of measured zero or “non-significant” *q* classified instead by the ANNs as “significant” (the contrary case is less interesting, since the original CLASH ANN tended to overestimate, and the classifier was introduced to overcome this issue). The scheme of Fig. 3.8 resumes the percentage of wrongly classified tests belonging to the single dataset of non-significant tests (please, note that in Verhaeghe et al., 2008, the case of “non-significant” overtopping is labeled as “class -1”, while the “significant” as “class +1”). As it can be appreciated from this figure, from the total amount of tests effectively belonging to “class -1”, the classifier ANN passes to the quantifier a 19.29% of tests, i.e. it wrongly classifies as “significant” only a 19.29% of tests. These tests (reduced to the 18.22% since a 1.07% presented input parameters out of the range of training of the quantifier ANN) are all processed by the quantifier ANN as “significant”, for a resulting 18.22% of wrong predictions (overestimations).

By comparing these results to the single quantifier CLASH ANN (grey captions, in Fig. 3.8), which definitely overestimates a 54.41% of tests, the two-phase ANN significantly reduces the number of overestimated values of *q* with respect to the “simply” quantifier CLASH ANN.

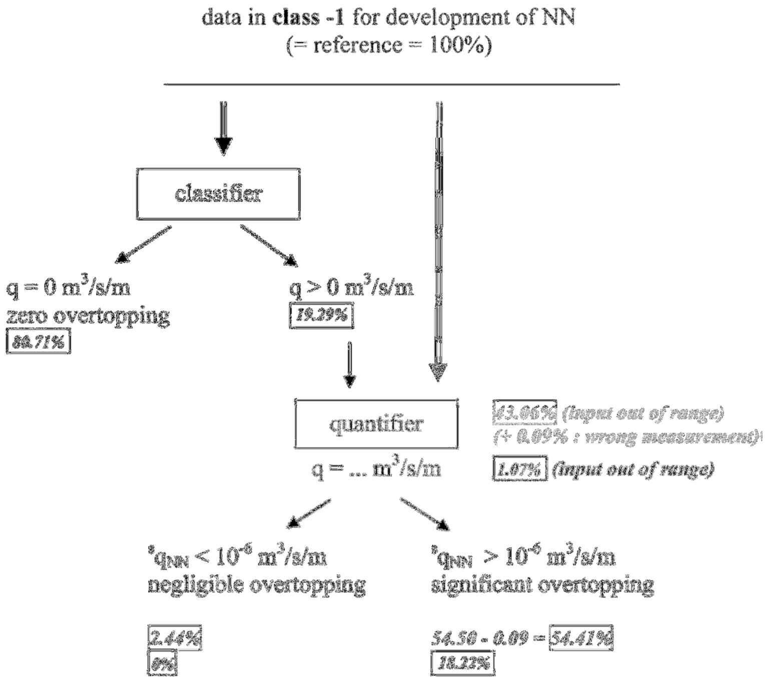


Figure 3.8 – Black colour: conceptual scheme and performance of the 2-phase ANN (Verhaeghe et al., 2008); grey colour: performance of the original CLASH ANN (Van Gent et al., 2007)

Therefore, the adoption of a two-phase classifier-quantifier ANN allowed a reduction of the overestimation errors, referring in particular to the cases of small and zero-overtopping, against a modest increase of the general performance function (*rmse*).

However, the ANN effectively accepted by the CLASH project was the one proposed by Van Gent et al. (2007). This ANN was preferred since a "classifier" – to be properly trained – would need an almost equal quantity of "significant" and "non-significant" tests, while the dataset including q>10^{-6} m^3/(sm) is much more extended (the proportion is approximately 4:1). This different distribution of data is principally due to two reasons: first, the experiments were obviously carried out in order to gather results of significant overtopping rather than non-significant or null overtopping; second, a great part of zero-overtopping tests was attributed a value of RF=4, and therefore discarded as "unreliable". Indeed, the cases of identically zero-overtopping are ambiguous, due to a non-homogeneous interpretation of the "zero" value during the several experiments: e.g., for some large-scale tests, it was attributed $q=0$ when values of q<10^{-3} m^3/(sm) occurred; for small-scale tests generally it was set q=0 when q<10^{-6} m^3/(sm), but in some cases even lower values of q, down to q<10^{-9}, were registered (Van Gent et al., 2007).

3.4 Prediction of the wave transmission coefficient behind low-crested structures

The "Low-Crested Structures" (LCS) alternatively work in submerged or emerged conditions according to the sea states. During the last twenty years, the LCS have been object of many investigations finalized to evaluate and model the induced wave energy dissipation, of which the wave transmission coefficient K_t is considered the main indicator. Indeed, for these structures referring to "wave transmission" is more correct than to "wave overtopping".

The quantification of the wave transmission/overtopping is more complicate in case of such structures, because of the variability of the working conditions. The most commonly adopted approach to design LCS is the employment of empirical formulae which provide an estimate of K_t on the basis of the physical parameters describing the geometric characteristics of the structures and the wave attack conditions.

Panizzo and Briganti (2007), starting from the traditional concept of "fitting" the experimental data, observed that the empirical formulae so far available in literature were affected by a considerable scatter and some shortcomings principally caused by discontinuities of the ranges of the involved physical parameters. Therefore, developing some preliminary results obtained by Panizzo et al. (2003), they created a numerical model for the prediction of K_t behind low-crested structures based on the neural networks.

The database they employed to train the ANN consisted of 2'285 tests overall, and was in a great part derived from the larger database collected within the EC project DELOS ("*Environmental design of low crested coastal defense structures*") which in turn was composed by data coming from several 2D tests on coastal structures and wave transmission performed in different European laboratories and facilities. Since the training database plays a key role for the ANN performance, Panizzo and Briganti investigated the distribution of the tests, by subdividing the database in seven homogeneous datasets according to the type of LCS. The ranges of variability of the physical parameters were analyzed for each of the seven datasets through frequency histograms.

The ANN was tested against each of these homogeneous datasets.

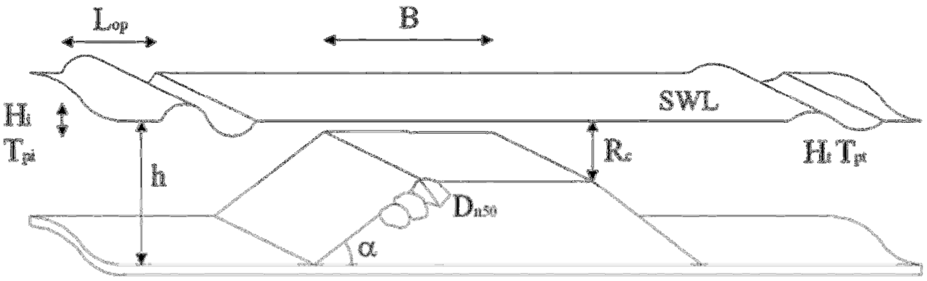


Figure 3.9 - Structural and hydraulic parameters involved in the definition of the ANN by Panizzo and Briganti (2007).

The construction of the ANN started with the definition of the input parameters and their re-scaling into the range [-1;+1], accounting for their minima and maxima. The number of the input parameters was set to 6, which are reported in the following:

- R_c/H_i, the relative structure's crest-freeboard;
- $H_i/D_{n,50}$, accounting for the structure permeability and describing the wave pressure;
- B/H_i, the non-dimensional structure's crest width with respect to the significant wave height;
- B/L_{op}, the non-dimensional structure's crest width with respect to the spectral wave length;
- $\xi_{o,p} = tan\alpha / \sqrt{\frac{H_i}{L_p}}$, the breaking parameter;
- H_i/h, representing the ratio between significant wave height and water depth.

The number of hidden neurons was set to 6, all included in one hidden layer. These features were defined after an optimization process similar to the one described by Van Gent et al. (2007) for the CLASH ANN (see Fig. 3.3). The output neurons was one, and coincided to K_t. The logical layout of such defined ANN is here reported in Fig. 3.10.

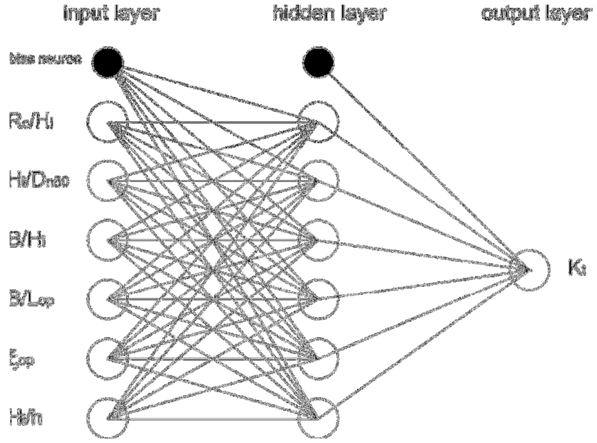


Figure 3.10 – Architecture of ANN by Panizzo and Briganti (2007) for the prediction of the wave transmission coefficient.

The activation of the hidden neurons was driven by a sigmoid transfer function, while for the output neuron, the activation function was a linear function (see, respectively, Eq.s 3.6 and 3.7).

$$TF(A_m)=\frac{1}{1+e^{-A_m}},$$

Eq. 3.6

$$TF(A_m)=A_m$$

Eq. 3.7

where $TF(A_m)$ is the activation function of the activation value A_m of the neuron m. The performance (or cost) function was defined as in Eq. 3.8:

$$E=\frac{1}{2}\sum_{p=1}^{P}\|Y-K_t\|^2,$$

Eq. 3.8

where Y is the ANN predicted value of K_t and K_t is the known experimental value; P is the total available number of data for K_t (then, the number of elements of the vectors Y and K_t)

The algorithm of Levenberg-Marquardt (Marquardt, 1963) was used to train the ANN and therefore defining the weight matrix (W) and the bias array (b). The algorithm minimizes the performance function (Eq. 3.9) by keeping as small as possible the step between the old and the updated configuration of W and b. For more details, see paragraph 2.3.5.3.

The adopted procedure to validate the ANN applicability and accuracy was based on the "early-stopping" technique (see paragraph 2.2.2). The database was subdivided into three parts: an 80% of the data was employed for the training, and the remaining 20% was used for the testing and validating sets, as required by the early-stopping. Since the order of magnitude of the computed error during the training and the testing phases were sensibly similar, Panizzo and Briganti concluded that the ANN was satisfactory accurate and could represent a reliable tool to predict the wave transmission within the field of validity, established by the ranges of the input parameters.

The goodness of the outcomes provided by the ANN was remarked by comparing its performance with the one of the "traditional" available predicting methods for K_t behind LCS. In particular, the ANN behavior was compared to the empirical relationships by van der Meer et al. (2005), which were calibrated on the same dataset (here reported in Eq. 3.9, where the symbolism of Fig. 3.9 is adopted).

$$K_t=-0.4\cdot\frac{R_c}{H_i}+0.64\left(\frac{B}{H_i}\right)^{-0.31}\cdot\left(1-0.5\xi_{op}\right),\ for\frac{B}{H_i}<10$$

$$K_t=-0.35\cdot\frac{R_c}{H_i}+0.51\left(\frac{B}{H_i}\right)^{-0,65}\cdot\left(1-0.41\xi_{op}\right),\ for\ \frac{B}{H_i}>10\ \text{and for submerged structures}$$

Eq. 3.9

The main results of the comparison is here reproduced in the plots of Fig. 3.11, where the ANN predictions of K_t as functions of the corresponding experimental values (plot to the left) are compared to the ones obtained by applying Eq. 3.9 on the same data (plot to the right). From Fig. 3.11, the improved accuracy associated to the ANN predictions is evident, by considering:

- the reduced scatter and the symmetry of the distribution of the predictions around the bisector representing the perfect conditions $K_{t,ANN} = K_{t,s}$;
- the narrower bands representing the 95% confidence intervals;
- above all, the overcoming of the upper-limit associated to Eq. 3.9, which prevent the prediction of values of K_t >0.85.

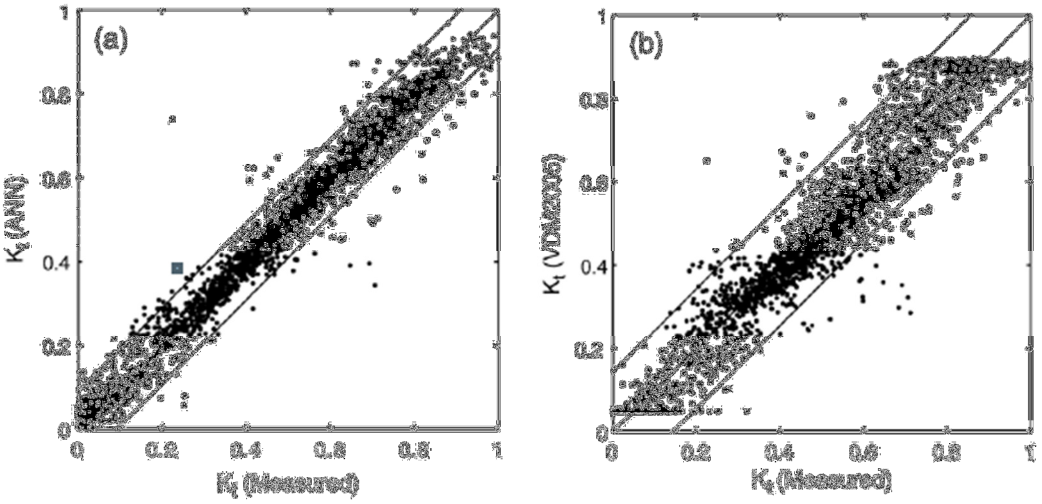


Figure 3.11 – Comparison among the results obtained from the ANN by Panizzo and Briganti (2007) and the formula by van der Meer et al. (2005) for the prediction of the wave transmission coefficient.

The better performance provided by the ANN can be explained by considering the larger number of the involved parameters and the adoption of non-linear functions to fit the experimental data.

Despite the promising results, the ANN by Panizzo and Briganti could be further improved by extending the range of validity of the ANN itself, e.g. by introducing the possibility to model also more geometrically complex structures or increasing the number of input parameters and hidden neurons.

3.5 Prediction of the wave reflection coefficient

Wave reflection from coastal and harbour structures may compromise structure stability, due to the induced intense scour at the structure toe, and may endanger harbour access, due to sea states at the entrance. Several empirical formulae for predicting the wave reflection coefficient K_r exist indeed, but most of them was developed on limited datasets related to specific structures (among the most recent studies, it could be mentioned: Muttray et al. (2006) for structures with armour units; Calabrese et al. (2008) for low crested structures; Zanuttigh and van der Meer (2008), who developed a formula for the prediction of K_r for various type of structures, emerged and submerged, with and without berms, permeable and impermeable; Zanuttigh and Lykke Andersen (2010), for oblique wave attacks.

Besides these empirical formulae, wave reflection can be estimated using numerical models. Depth integrated models based on both Non-Linear Shallow Water Equations and Boussinesq-type equations were used to assess wave transformations induced by coastal structures (e.g. Wurjanto and Kobayashi, 1993, Johnson et al., 2005). For a near-field analysis, depth-resolving models are preferred. Recently, solvers based on the Reynolds Averaged Navier Stokes and a Volume of Fluids technique for the free surface tracking (RANS-VOF) models have been used (Lara et al., 2006; Losada et al., 2008). Notwithstanding faster computing resources, they are still time-consuming and may lead to overestimation of wave reflection (Zanuttigh et al., 2010).

In alternative to traditional techniques, ANNs offer flexibility and accuracy. From this background, Zanuttigh et al. (2013) realized an ANN model for the prediction of K_r. It was trained on a database of nearly 5’800 data, including structures with straight and non-straight slopes; seawalls, caissons and circular caissons; aquareefs and structures under oblique wave attacks. A large part of the employed database corresponded to the original wave reflection database by Zanuttigh and van der Meer (2008), which, in turn, derived from the wave overtopping database of the CLASH project (van der Meer et al., 2008). The format of the database follows precisely the structure of the wave overtopping database gathered within the CLASH project.

The final form of the ANN architecture is portrayed in Fig. 4.12. It consisted of 13 input elements and 40 hidden neurons; the output neuron was naturally K_r. It was built in the *Matlab* environment and is characterized by the following fundamental characteristics:

- multilayer network, based on a “feed-forward back-propagation” learning algorithm;
- static network (absence of delays and feedbacks);
- hidden neurons transfer function: hyperbolic tangent sigmoid transfer function;
- output neuron transfer function: linear transfer function;
- training style: “batch training”, connections weights and biases are updated at the end of each training epoch, just once the ANN has read all the input data;
- training algorithm: Levenberg-Marquardt algorithm (Marquardt, 1963);
- learning algorithm: momentum gradient descent back-propagation algorithm;
- generalization technique: early-stopping.

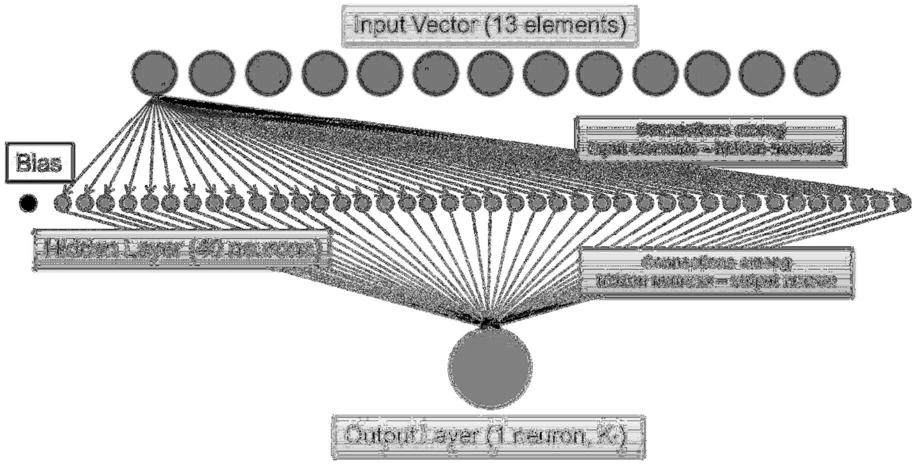

Input Vector (13 elements)
Connections among
input elements – hidden neurons
Bias
Hidden Layer (40 neurons)
Connections among
hidden neurons – output neuron
Output Layer (1 neuron, K_r)

Figure 3.12 – Architecture of the ANN by Zanuttigh et al. (2013) for the prediction of the wave reflection coefficient.

Each of the 13 input elements, similarly to the ANN for the wave transmission by Panizzo and Briganti (2007), was represented by a non-dimensional parameter reproducing a specific physical quantity or process. They are listed in the following (for the adopted symbolism, please refer to Fig. 3.1):

- $H_{m,0,t}/L_{m-1,0,t}$: it is proportional to the wave steepness and it is part of the breaking parameter, which represents the breaker level of energy.
- $h_t/L_{m-1,0,t}$:it accounts for shoaling effects associated to incident waves;
- $cot\alpha_d$: off-shore structure slope in the run-up area; together with (1), it completes the description of $\xi_{0,p}$. The cotangent form was preferred to the tangent to prevent infinite values of the tangent in presence of seawalls, i.e. when $\alpha_d = 90°$;
- γ_f: armour layer roughness factor. It is an index of wave energy dissipation during the run-up process. The higher the values of γ_f, the smoother and the more reflective the structure.
- $R_c/H_{m,0,t}$: the lower the relative crest freeboard, the greater the wave transmission and the lower the K_r.
- $D_{n,50}/H_{m,0,t}$: this term essentially represents the wave pressure inside the structure pores; it is involved in the definition both of design conditions and of the ranges of validity for existing K_r formulae (Davidson et al., 1996; Zanuttigh and van der Meer, 2008; Calabrese et al., 2008). In addition, Panizzo and Briganti (2007) used this parameter as input element for ANN they developed to predict the values of the transmission coefficient for LCSs.
- β: this parameter is essential to describe the effects of oblique wave attacks. The greater the wave obliquity the greater the structure surface exposed to wave action and therefore the greater the wave dissipation and the lower the wave reflection. Usually, the reductions of wave run-up and wave overtopping due to wave obliquity are represented by the factor γ_β (EurOtop, 2007), that depends on β through coefficients still needing calibration. Zanuttigh and Lykke Andersen (2010) had already showed the significance of γ_β for predicting K_r. In order to avoid the use of coefficients that may be inaccurate, it was selected to use directly β rather than γ_β.
- $G_c/L_{m-1,0,t}$: this parameter was introduced to represent any possible reflected wave phase delay from the upper part of the breakwater.
- $B/L_{m-1,0,t}$: this parameter, as well the following $h_B/H_{m,0,t}$ and $cot\alpha_{incl}$, was used to describe the effects induced by a berm. It is conceptually similar to $G_c/L_{m-1,0,t}$, in order to

account for the phase delay of waves reflected from the structure down-slope and from the berm.

- $h_B/H_{m,0,t}$: this parameter is essentially a modified breaking index to account for the process of waves breaking on the berm, dissipating energy and reducing the run-up process over the structure upper slope.
- $cot\alpha_{incl}$: also the knowledge of the angle α_{incl} is important for the network as long as it differs from α_d, i.e. when the structure has a berm, because of the difference in estimating ξ_0.
- m: a foreshore in front of a breakwater might induce additional reflection while waves travel from offshore to the structure toe.
- *spreading*: the directional wave spreading tends to increase the effects induced by wave obliquity, and therefore the greater the directional spreading the lower the wave reflection.

The performance of the ANN was qualitatively shown by the comparison of the predicted values of $K_{r,ANN}$ with the measured ones $K_{r,s}$ (here reported in Fig. 3.13, left) and by the dispersion of the difference $e = K_{r,s} - K_{r,ANN}$ as a function of $K_{r,s}$ (here Fig. 3.13, right). In Fig. 3.13, left, the central line represents the bisector, i.e. the perfect correspondence among predicted and experimental values, while the external lines represent the 95% confidence boundaries. This kind of diagram is completely similar to the ones already plotted by both Van Gent et al. (2007) and Panizzo and Briganti (2007), for, respectively q (see Fig. 3.4) and K_t (see Fig. 3.11).

Both the graphs of Fig. 3.13 show the good agreement of computations and measurements; the confidence interval is quite narrow and, moreover, the ANN appears to provide the results with a good degree of symmetry.

An important novelty which can be found in the work by Zanuttigh et al. (2013) is the plotting of the distribution of the error $e = K_{r,s} - K_{r,ANN}$ through a frequency histogram (here Fig. 3.14). The form of the histogram not only demonstrated that the error distribution was symmetric and pretty narrow, but also that it followed a Gaussian curve, a paramount issue which allows to employ the standard deviations (and the percentiles 95% displayed in Fig. 3.13) as indicators of the ANN performance.

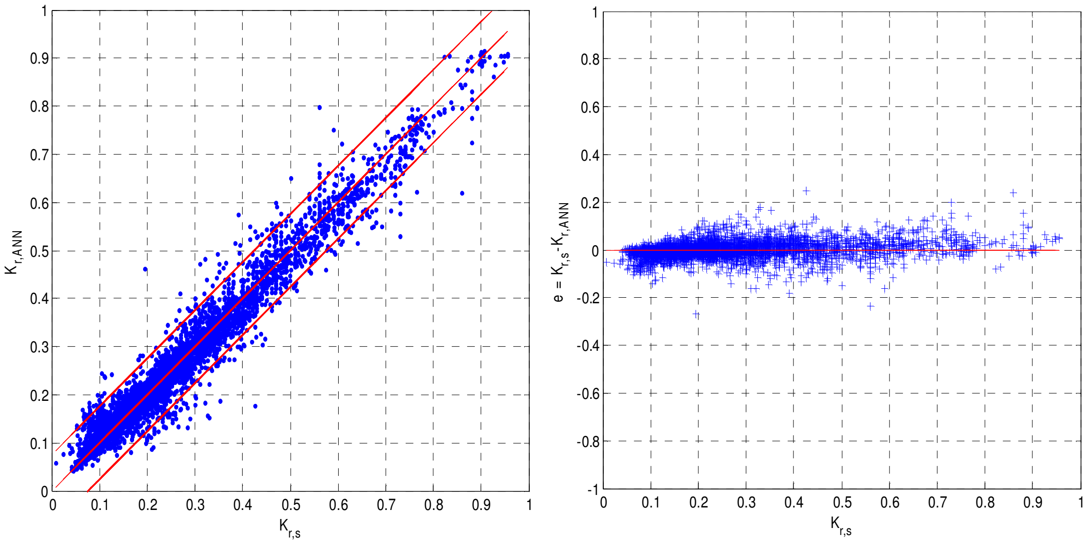


Figure 3.13 – Left: comparison among K_r predicted values ($K_{r,ANN}$, ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); on the right: difference $e = K_{r,s} - K_{r,ANN}$ (ordinate) as a function of $K_{r,s}$ (abscissa).

The observed Gaussian distribution justifies the "committee of networks" (see paragraph 2.2.3) as technique to contemporary improve and assess the ANN performance. In other words, providing the user with an "average" ANN and "average" predictions – or a distribution of quantiles – is allowed only if it is proved that the error computed by the ANN follows a Gaussian distribution. The results by Zanuttigh et al. (2013) demonstrated it, at least for what concerns the wave reflection.

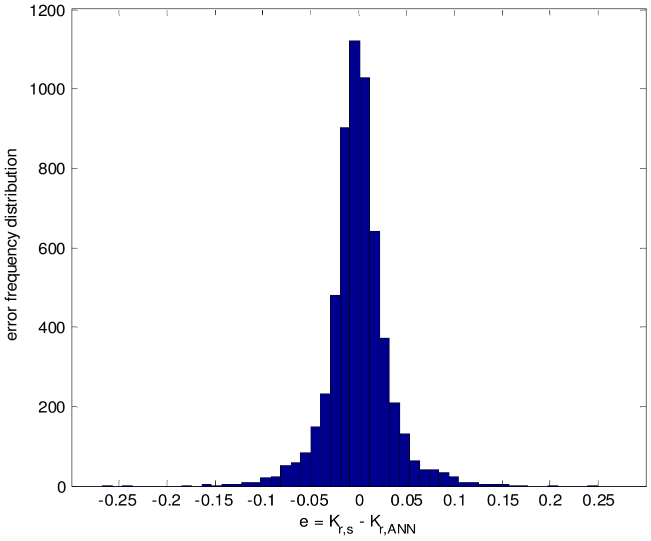


Figure 3.14 – Frequency distribution histogram of the difference $e = K_{r,s}$ - $K_{r,ANN}$.

Zanuttigh et al. (2013) discussed the quantitative performance of the reflection ANN through the computation of the mean and standard deviation values of three error indexes derived from 40 simulations. The three error indexes were the *rmse*, the Willmott index *WI* (Willmott, 1981) and the coefficient of determination R^2. Each of them (whose definition is later reported in Eq.s 4.3, 4.4 and 4.5) describes a different aspect of the error distribution (as remarked in details in paragraph 4.3). In particular, R^2 and *WI* present the advantage of being non-dimensional, therefore they do not depend on the measure of the data and are comparable among different applications and works. For these reasons, the same three indicators have been used also to evaluate the performance of the ANN presented and described in this thesis (see Chapter 5).

It was proved that the ANN predictions were particularly accurate (rmse = 0.038±0.003), especially if compared to existing formulae, characterized by larger values of rmse and associated to well defined structures typologies and therefore more restricted datasets (Zanuttigh and van der Meer, 2008; Zanuttigh and Lykke Andersen, 2010). The values of the standard deviations of rmse, WI and R2 were all around 10-3, denoting a good ANN stability, in spite of the random data-selection processes (training and testing).

The uncertainty of the predictions was analyzed through the technique of bootstrap resampling the database. 500 resamples of the training set were performed and the corresponding predictions computed. The average results show almost the same error distribution as the ones obtained from the non-bootstrapped ANN, demonstrating a good stability of the model.

Further research should be performed to address the accurate representation of more complex geometries in the ANN, such as breakwaters or vertical walls with double crown walls or combined crown walls (i.e. crown walls composed by different slopes), breakwaters with storage basins on the crest, perforated and Jarlan-type caisson.

4. A NEW ARTIFICIAL NEURAL NETWORK FOR THE PREDICTION OF THE WAVE OVERTOPPING DISCHARGE, THE WAVE REFLECTION AND THE WAVE TRANSMISSION COEFFICIENTS

4.1 Introduction

The design of coastal and harbour structures requires a systematic analysis of all the processes of wave-structure interaction, which takes into account the combined effects of wave overtopping, wave transmission and wave reflection. Indeed, all these phenomena should be considered as different outcomes of the same physical process, and therefore should be investigated contemporarily.

The development and/or use of an Artificial Neural Network (ANN) is therefore particularly recommended in case of complicated structure geometries and variable wave conditions. This kind of predictive method requires however a homogeneous and "wide-enough" database: the number of data should be sufficient for training the ANN based on a number of ANN parameters and including a sufficiently wide number of data for all range of possible output values. There are specific cases that also an ANN cannot deal with, such as very complex walls, perforated caissons and double promenades, see for details EurOtop (2007) and specifically the methodology released within the PC-OVERTOPPING calculator (http://www.overtopping-manual.com/calculation_tool.html).

All the existing ANNs (see Chapter 3) actually proved to be able to overcome some of the limits imposed by the traditional empirical formulae, but each of them is still restricted to reproduce only one of the processes involved in the wave-structure interaction. Nevertheless, the assumption that all the processes are physically correlated implies that a unique set of physically based parameters can be defined to represent all the phenomena.

Therefore, aim of this work is to further develop the existing ANNs in order to deliver a tool which is able to estimate the wave overtopping discharge (q), the wave transmission and the wave reflection coefficients (K_r and K_t) at once, i.e. by means of just one set of input parameters and of the same ANN architecture.

This Chapter is dedicated to the presentation of such a tool, focusing on the detailed description of its input elements and the main features of its internal architecture (respectively, Paragraphs 4.3 and 4.4). This advanced ANN has been developed after a step-by-step optimization process started from a preliminary ANN for the evaluation of the wave reflection coefficient (Zanuttigh et al., 2013). The preliminary ANN was then applied to the prediction of the wave transmission coefficient (Formentin and Zanuttigh, 2013) and, in a later moment, of the wave overtopping discharge (Zanuttigh et al., 2014). The process has concluded with a final revision of the input parameters and architecture features, which has leaded to the realization of a single ANN model able to predict all the three wave-interaction phenomena introduced above.

Since the basis of a "good" ANN is essentially the database to be used for training, the first step consisted in the arrangement of a "wide-enough" and homogeneous collection of tests, preferably organized following the same structure schematization. Indeed, the data, the set-up and the structure parameters of CLASH (2004) were used as a starting point to gather the other tests and organize them according to the same approach. The resulting final database is described in details in Paragraph 4.2.

The choice of the best ANN layout represented the second step of the work. The overall ANN "layout" accounts of both the input parameters and the internal architecture. To define and calibrate these elements, two of the existing ANNs were analyzed, tested and modified, specifically the original CLASH ANN (Van Gent et al., 2007) and the preliminary ANN for reflection previously developed by Zanuttigh et al. (2013).

The process of selection and optimization of the input parameters, in comparison to these ANNs is reported in Paragraph 4.3, while a separate section (Paragraph 4.4) illustrates the ANN architecture. This section includes a synthesis of the methodology followed to define the most relevant elements of the architecture itself, from the number of hidden neurons to the several elements characterizing the training phase. In particular, since the existing ANNs propose different methods for the performance assessment, the employment of different techniques and its fallout on the valuation of the performance is discussed. Furthermore, the adoption of the bootstrap resampling technique and the effective optimal number of required resamples is described.

A final paragraph is dedicated to the analysis of performance of the preliminary ANN against the prediction of the wave reflection and the wave transmission (Paragraph 4.5). Aim of this section is to present the steps and the followed approach to optimize the layout of the final new ANN and assessing its performance. The methodology and the sensitivity analysis carried out to calibrate the preliminary ANN are essentially the same adopted for the new ANN.

The results of the new ANN and the discussion about its qualities and shortcomings are gathered in the next Chapter 5.

4.2 The database

A homogenous database of 16'165 tests has been gathered, starting from the original CLASH database (van der Meer et al., 2008) and extending it in order to include other wave overtopping, reflection and transmission tests. The assemblage of the data has been carried out by keeping the set-up of the original database, i.e. by following the same schematization of the structures (see Fig. 4.1) and by maintaining the same geometric parameters, as well as the relevant climate parameters, already identified within CLASH project.

In addition, with respect to the original CLASH database:

- the values of K_r and K_t have been included where available;
- an additional geometric parameter has been introduced: the average unit size D representative of the structure elements, which has proved to be relevant in the prediction of K_r and K_t (Panizzo and Briganti, 2007; Formentin and Zanuttigh, 2013; Zanuttigh et al., 2013);

Therefore, the final "extended" database, in its final layout, consists of:

- 14 hydraulic parameters, characterizing the wave attack conditions;
- 18 structural parameters, for the as general as possible description of the cross-section of the structures;

- general parameters, the reliability and the complexity factors, the identify label of the test and the identifying marks of the armour unit/type.

Table 4.1 reports the type and the number of all the parameters included in the extended database, in comparison to the original CLASH database, while the most relevant parameters are sketched in Figure 4.1; for a more detailed description, see van der Meer et al. (2009).

Table 4.1 - Parameters included in the "new" extended database compared with the ones included in the original CLASH database.

#	Parameter	Unit of measure	Type	CLASH Database	Extended Database
1	*Name*	[-]	general	√	√
2	$H_{m,0,deep}$	[m]	hydraulic	√	√
3	$T_{p,deep}$	[s]	hydraulic	√	√
4	$T_{m,deep}$	[s]	hydraulic	√	√
5	$T_{m-1,deep}$	[s]	hydraulic	√	√
6	h_{deep}	[m]	structural	√	√
7	m	[-]	structural	√	√
8	β	[°]	hydraulic	√	√
9	*Spreading*	[-]	hydraulic		√
10	h	[m]	structural	√	√
11	$H_{m,0,t}$	[m]	hydraulic	√	√
12	$T_{p,t}$	[s]	hydraulic	√	√
13	$T_{m,t}$	[s]	hydraulic	√	√
14	$T_{m-1,t}$	[s]	hydraulic	√	√
15	h_t	[m]	structural	√	√
16	B_t	[m]	structural	√	√
17	γ_f	[-]	structural	√	√
18	$D_{n,50}$	[m]	structural		√
19	*Armour unit*	[-]	general	√	√
20	$cot\alpha_d$	[-]	structural	√	√
21	$cot\alpha_u$	[-]	structural	√	√
22	$cot\alpha_{excl}$	[-]	structural	√	√
23	$cot\alpha_{incl}$	[-]	structural	√	√
24	R_c	[m]	structural	√	√
25	B	[m]	structural	√	√
26	h_b	[m]	structural	√	√
27	$tan\alpha_b$	[-]	structural	√	√
28	B_h	[m]	structural	√	√
29	A_c	[m]	structural	√	√
30	G_c	[m]	structural	√	√
31	*RF*	[-]	general	√	√
32	*CF*	[-]	general	√	√
33	q	[m^3/sm]	hydraulic	√	√
34	*Pow*	[-]	hydraulic	√	√
35	K_r	[-]	hydraulic		√
36	K_t	[-]	hydraulic		√

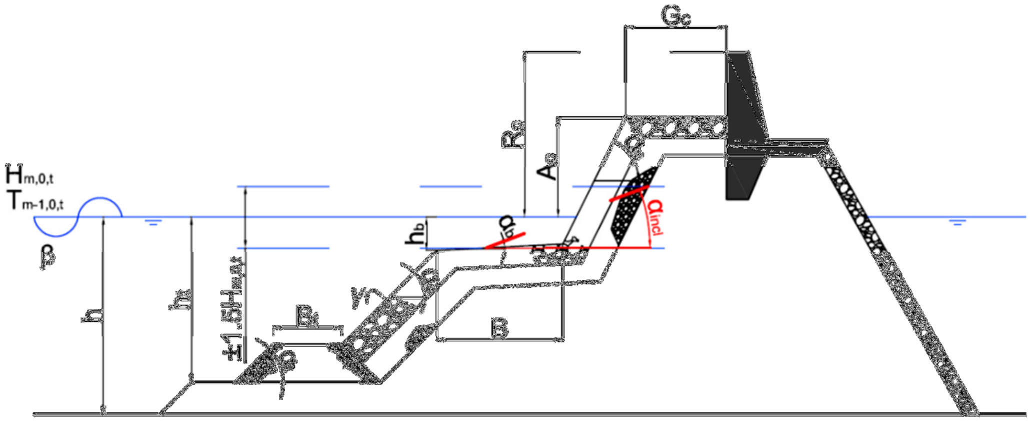


Figure 4.1 - Structure schematization including the hydraulic and structural parameters, based on CLASH terminology.

The complete database is organized into 7 “sections”, labeled progressively from A to G, in order to distinguish the different type of structures and wave attack conditions: rock permeable straight slopes (group “A”), rock impermeable straight slopes (group “B”), armour units straight slopes (“C”), smooth and straight slopes (“D”), structures with combined slopes and berms (“E”), seawalls (“F”) and oblique wave attacks (“G”).

Following this partition, the database can be analyzed considering the available data for each of the specific processes.

4.2.1 The wave overtopping database

Concerning the wave overtopping discharge q, a total amount of 11’825 tests are available. Besides the data collected from the CLASH database (van der Meer et al., 2009) – which consisted of more than 10’000 irregular wave overtopping tests and includes dikes, rubble mound breakwaters, berm breakwaters, caisson structures and combinations with complicated geometries – additional data on smooth steep slopes (Victor, 2012) and additional reports from LWI (Oumeraci et al., 2001, 2004 and 2007) have been added.

Figure 4.2 displays the distribution of the total amount of overtopping data within the seven “sections” from A to G. From this diagram, it is evident that the greatest part of the tests belongs to sections E and F.

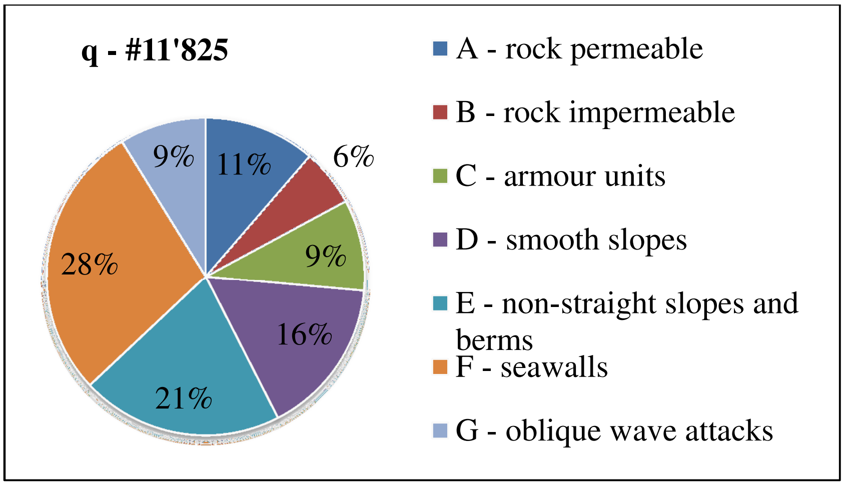


Figure 4.2 – Pie chart representing the distribution of the data within the wave overtopping database.

Since the CLASH database constitutes more than the 90% of the "extended" wave overtopping database, and since the CLASH database was already described in details in van der Meer et al., 2009, within this work no further information is reported.

4.2.2 The wave reflection database

Concerning the wave reflection coefficient K_r, 7'413 data are available. These tests are derived from the extended wave reflection database (Zanuttigh et al., 2013), which collects more than 5,700 data, including the original wave reflection database based on CLASH database (Zanuttigh and van der Meer, 2008) and additional data on seawalls (Oumeraci et al., 2001, 2004 and 2007), steep slopes (Victor, 2012) and berm breakwaters (Lykke Andersen, 2006).

Similarly to Fig. 4.2, the pie-chart diagram of Fig. 7.3 reports the distribution of reflection data on the seven sections from A to G. In this case, the widest sections are A, C and E, while the percentage of tests belonging to section F is sensibly more modest with respect to the wave overtopping database.

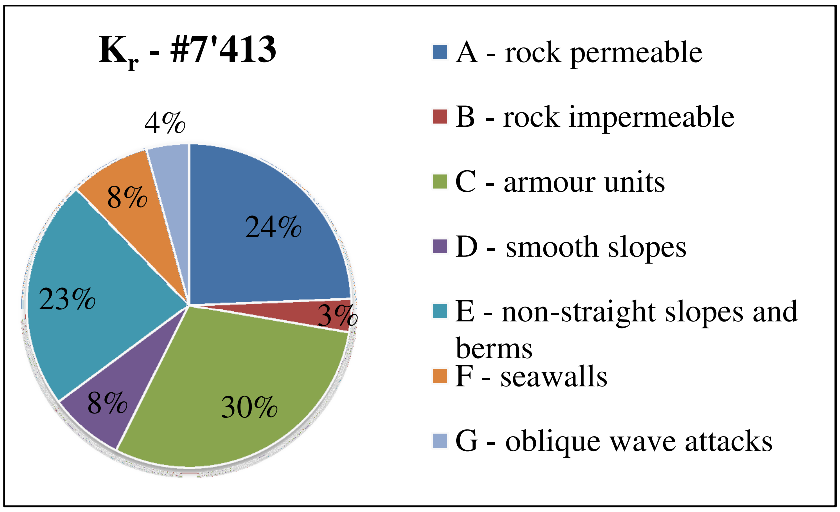


Figure 4.3 – Pie chart representing the distribution of the data within the wave reflection database.

Differently from the overtopping application, the description of the data present in the wave reflection database and the corresponding references is reported in details in Tab. 4.2.

Table 4.2 – Wave reflection database overview and references. Where the reference is not provided, the data were derived from private communications.

Database section	Structure type	Samples	References
A – Rock permeable straight slopes	Breakwaters	517	van der Meer (1988), Allsop and Channel (1989), Pearson et al. (2004), Davidson et al. (1996), Lykke Andersen and Burcharth (2004)
	LCSs	830	Seabrook and Hall (1998), Ruol et al. (2004), van der Meer et al. (2005), Zanuttigh and Lamberti (2006), Cappietti et al. (2006)
B – Rock impermeable straight slopes	Breakwaters	198	van der Meer (1988)
C – Armour units straight slopes	Accropods	26	Pearson et al. (2004)
	Antifer	29	Pearson et al. (2004)
	Tetrapods	29	Pearson et al. (2004)
	X-blocks	25	Pearson et al. (2004)
	Core-Locs	145	Pearson et al. (2004), Melito and Melby (2002)
	Cubes	234	Pearson et al. (2004)
	Haros	30	Pearson et al. (2004)
	Dolos	12	Pearson et al. (2004)
	Wave-walker blocks	63	
	Acquareefs	1063	Hirose et al. (2000)
D – Smooth straight slopes	Breakwaters	189	Pearson et al. (2004)
	T-blocks	23	
	Vilvord and Haringman blocks	17	
E – Structure with combined slopes and berms	Rocks	99	Lissev (1993)
	Rocks and asphalt	5	
	Tetrapods	50	
	Antifer	34	
	Smooth	265	
	Block revetments	57	
	Vilvord rocks and basalt	8	
	Smooth composite slope with vertical wall	170	Oumeraci et al. (2007)
F - SeaWalls	Vertical wall	27	Oumeraci et al. (2001)
	Caisson	5	
	Caissons and circular caissons	107	
	Vertical porous breakwater	59	Requejo et al. (2002)
G – Oblique attacks	Rocks LCS	84	Van der Meer et al. (2003)
	Large crested rocks	312	Lyke Andersen and Burcharth (2004)
	Large crested rocks	183	Lykke Andersen and Burcharth (2009)
	Short crested rocks	233	Lykke Andersen and Burcharth (2009)
	Cubes	249	Pearson et al. (2004)
	Long crested cubes	140	Lykke Andersen and Burcharth (2009)
	Short crested cubes	180	Lykke Andersen and Burcharth (2009)
	Smooth LCS	84	van der Meer et al. (2003)

4.2.3 The wave transmission database

Concerning the wave transmission coefficient K_t, additional data structures with berms (Lissev, 1993) have been combined with the DELOS database on low crested breakwaters (van der Meer et al., 2005; Panizzo and Briganti 2007), achieving a total amount of 3’366 test on rubble mound structures, aquareefs, rubble mounds with different kind of armour units and smooth slopes.

The diagram of Fig. 4.4 represents the distribution of the wave transmission data included within each of the seven sections.

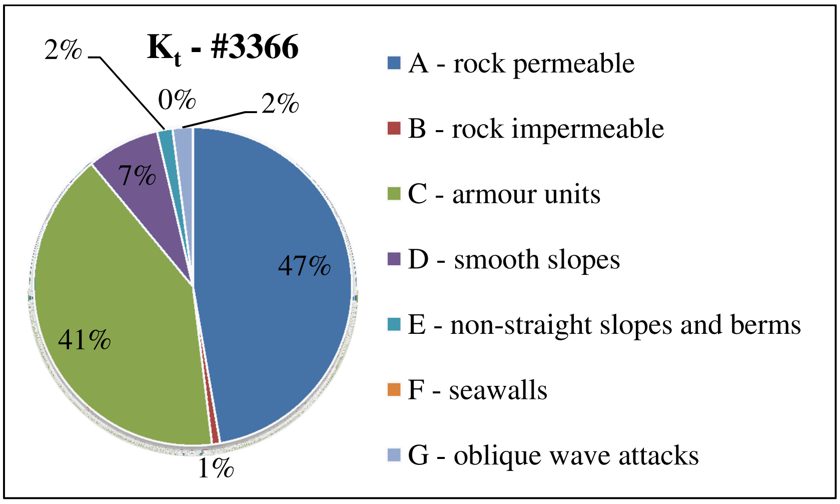


Figure 4.4 - Pie chart representing the distribution of the data within the wave transmission database.

Similarly to the wave reflection application, the description of the wave transmission tests and the corresponding references are detailed in Tab. 4.3.

Table 4.3 – Wave transmission database overview and references. Where the reference is not provided, the data were derived from private communications.

Database section	Structure type	Samples	References
A – Rock and rubble mound	Rock permeable	737	Seabrook and Hall, 1998; van der Meer, 1988; Daemen, 1991; Allsop, 1983
	Multilayer rock permeable	11	Ruol et al., 2004
	Rubble mound	460	Gironella et al., 2002; Garcia et al., 2004; Powell and Allsop, 1985; Calabrese et al., 2002; Ahrens, 1987; Daemrich et al., 2001
	Rubble mound	253	
B – Rock impermeable	Rock impermeable	69	Seelig, 1980
C – Armour units	Aquareef	1062	Hirose et al., 2002
	Core-Locs	122	Melito and Melby, 2002
	Tetrapods	196	Daemrich and Kahle, 1985

D+E - Smooth straight slopes and smooth berms	Smooth	215	Seelig, 1980;
			Daemrich and Kahle, 1985;

G – Oblique and 3D wave attacks	Rock oblique	170	van der Meer et al., 2003; Kramer et al., 2005;
	Smooth oblique	84	van der Meer et al., 2003

4.2.4 Discussion about the distribution of data in the extended database

By comparing the pie charts relative to the three databases of wave overtopping (Fig. 4.2), reflection (Fig. 4.3) and transmission (Fig. 4.4), it is evident that the distribution of the tests is very different. For example, some typology of tests such as the structures with berms and seawalls are completely absent from the transmission database, while the reflection database, as well as the overtopping database, includes all the different kind of tests.

Furthermore, the three databases are only partially overlapped, i.e. only some selected datasets from each database contemporarily include more than one of the outputs parameters (q, K_r and K_t), and specifically:

- the value of q and K_r is contemporarily available for 2'065 tests, which represent approximately the 13% of the total data, the 28% of the reflection tests and the 17% of the overtopping tests;
- the value of K_r and K_t is contemporarily available for 2'303 tests, i.e. approximately the 14% of the total data, the 31% of the reflection tests and the 68% of the transmission tests;
- the percentage of tests for which both q and K_t are known is insignificant (less than 1%), and therefore also the percentage of tests for which all the output parameters are available is too limited.

These percentages are far limited to allow a "contemporary" prediction of more output parameters. In other words, the possibility to train an ANN whose output layer includes all the three parameters has to be excluded. The actual unfeasibility of such possibility is discussed and proved within a dedicated section of the present work (Paragraph 5.4). Depending on the possibility to gather more data on combined measurements, the development of contemporary predictions is postponed to future research.

At the contrary, the purpose to build an ANN characterized by a same architecture and a same input set which can describe all the three processes can be attended.

4.3 New ANN input parameters

This section presents the optimized input set selected for the new ANN and some results of the sensitivity analysis carried out to define it. The final 15 non-dimensional parameters are reported in Tab. 4.4. This table shows also the input elements characterizing two of the existing ANNs, adopted to perform the comparison with the new ANN. These two ANNs are:

- the ANN for wave overtopping developed within the CLASH project by Van Gent (2007), hereafter ANN(1);
- the ANN optimized for the wave reflection proposed by Zanuttigh et al. (2013), named in the following ANN(2).

The ANN(1) was optimized for the evaluation of q values greater than 10^{-6} m^3/(sm) and it was originally trained against the original CLASH database (van der Meer et al., 2009). The experimental data were scaled according to the Froude Law by dividing each quantity with the significant wave height, $H_{m,0,t}$,. Therefore the data were basically rescaled to a same prototype condition ($H_{m,0,t}$, = 1 m). It included 15 "dimensional" input parameters, describing both geometric and climate characteristics. In addition, the output parameter q was scaled and transformed in logarithmic scale, before applying the training process.

The ANN (2) was developed for the prediction of K_r.(see Paragraph 3.5). It was trained against an extended version of the CLASH database, which included additional datasets on impermeable slopes, seawalls and 3D wave attack. The input vector comprehended 13 non-dimensional elements (Zanuttigh et al., 2013), reproducing a specific physical parameter or process, for example: $H_{m0,t}/L_{m-1,0,t}$ represents the wave steepness, $h_t/L_{m-1,0,t}$ the shoaling index, etc. To this purpose, this ANN (2) used two scaling parameters, i.e. a depth and a width, respectively $H_{m0,t}$ and $L_{m-1,0,t}$, instead of the Froude scaling as ANN (1).

The input set of the new "optimized" ANN has been defined through a sensitivity analysis, selection and revision of the input parameters of both ANN (1) and ANN (2). As stated above, Tab. 4.4 compares the input parameters of the new ANN to the ones of the existing ANNs. The CLASH symbols and terminology (see Fig. 4.1) are adopted.

Table 4.4 - Input parameters of the three tested ANNs: the original CLASH ANN, the ANN for the prediction of the wave reflection developed by Zanuttigh et al. (2013) and the "new" optimized ANN. The total number of 14 parameters for the W.R. ANN includes the wave period $T_{m-1,0,t}$ and the water depth h in front of the structure implied in the calculation of the wave length $L_{m-1,0,t}$; the 16 parameters for the optimized ANN includes the structure freeboard $A_c/H_{m,0,t}$ which is implicitly involved in the definition of the crest freeboard $R_c/H_{m,0,t}$.

#	ANN(1) (CLASH)	ANN(2) (Zan. 2013)	New ANN
1	$H_{m,0,t}$	$H_{m,0,t}/L_{m-1,0,t}$	$H_{m,0,t}/L_{m-1,0,t}$
2	$T_{m-1,t}$	$h_t/L_{m-1,0,t}$	$h/L_{m-1,0,t}$
3	γ_f	γ_f	$h/H_{m,0,t}$
4	$\cot\alpha_d$	$\cot\alpha_d$	γ_f
5	$\cot\alpha_u$	$\cot\alpha_{incl}$	$\cot\alpha_d$
6	B	$D/H_{m,0,t}$	$\cot\alpha_{incl}$
7	B_t	$R_c/H_{m,0,t}$	$D/H_{m,0,t}$
8	h	$B/L_{m-1,0,t}$	$R_c/H_{m,0,t}$
9	h_t	$h_b/H_{m,0,t}$	$B/L_{m-1,0,t}$
10	h_b	$G_c/L_{m-1,0,t}$	$h_b/H_{m,0,t}$
11	R_c	m	$G_c/L_{m-1,0,t}$
12	A_c	β [rad]	m
13	G_c	Spreading	β [rad]
14	$\tan\alpha_B$		$B_t/L_{m-1,0,t}$
15	β		$h_t/H_{m,0,t}$
16			
TOT	15	14	16
Data required to train the ANN	**762**	**722**	**802**

From Table 4.4, it can be observed that for the optimized:

- the "physically-based" scaling methodology of ANN (2) – with $H_{m,0,t}$ and $L_{m-1,0,t}$ as scale parameters – has been kept;
- the inputs B_t and A_c from ANN (1) (not present in ANN (2)) have been kept; though A_c does not directly compare within the input set, it is indeed included in the definition of $R_c/H_{m,0,t}$, as depicted in Eq. 4.1:

$$R_c/H_{m,0,t} = max(R_c/H_{m,0,t},\ A_c/H_{m,0,t});$$

Eq. 4.1

Such a definition is justified by the observation that only a few types of structures in the database actually present $A_c > R_c$ (i.e. the ones characterized by a promenade behind the crown wall), and for these specific cases the highest crest is the parameter to determine the wave overtopping discharge falling inside the promenade.

- since the inputs D and m have demonstrated to represent key parameters for the wave reflection and wave transmission (Formentin et al., 2013), they have been kept from ANN(2);
- the input from ANN(2) $\cot\alpha_{\mathrm{incl}}$ representing the "average" slope has been preferred to the input from ANN(1) $\cot\alpha_{\mathrm{u}}$ representing the "upstream" slope, since this choice allows to eliminate the two parameters $\tan\alpha_{\mathrm{B}}$ and B_h and consequently reduce the risk of over-fitting (see Paragraph 2.2.2).
- an overall revision of the input parameter $h_t/L_{m-1,0,t}$ from ANN(2) has leaded to the definition of the three parameters $h/L_{m-1,0,t}$, $h/H_{m,0,\mathrm{t}}$ and $h_t/H_{m,0,t}$. The single parameter $h_t/L_{m-1,0,t}$ was the most suitable for wave reflection and transmission (Formentin et al., 2013), but it has been substituted in the final three-parameters-combination following a specific sensitivity analysis which accounted also for the wave overtopping.

Table 4.4 also reports a last line that summarizes the minimum required number of tests to train each ANN; this value strictly depends on the number of input parameters and hidden neurons, and it exactly corresponds to the total number of connections among input parameters, hidden and output neurons and biases (see Fig. 4.3 and Paragraph 4.4.2). Consequently, the minimum required of tests, N_{tests}, can be easily determined as stated in Eq. 4.2.

$$N_{tests}=HN\cdot(I+2)+ON\cdot HN+b\cdot HN+(b+1)\cdot ON,$$

Eq. 4.2

Where: HN = number of hidden neurons; I = number of input parameters; ON = number of output neurons; b = bias.

Within Tab. 4.4, the minimum amount of tests is computed considering the optimized number of 40 hidden neurons (see Fig. 4.8 and Paragraph 4.4.2) and the total number of "actual" input parameters. Indeed, both ANN (2) and the new "optimized" ANN implicitly require an additional parameter:

- for ANN (2), the computation of the wave length $L_{m-1,0,t}$ requires both the values of the experimental spectral wave period $T_{m-1,0,t}$ and water depth h;
- for the "optimized" ANN, the definition of the relative crest freeboard $R_c/H_{m,0,t}$, involves the parameter $A_c/H_{m,0,t}$ (see Eq. 4.1).

In order to distinguish among the effective input elements belonging to the input vector of the new ANN (Tab. 4.3) and the quantities involved in the definition of the input elements themselves, these latter are reported in Tab. 4.5.

Table 4.5 –Parameters required for the definition of the input vector elements of the new "optimized" ANN.

Wave attack	Slope	Toe	Berm	Crest and crown wall	Roughness/ Element size
$H_{m0,t}$	$cot\alpha_d$	B_t	B	A_c	γ_f
$T_{m-1,0,t}$	$cot\alpha_{incl}$	h_t	h_B	G_c	D
β	m			R_c	
h					

From Tab. 4.4, it is clear that the new ANN requires a greater number of training tests, with respect to both ANN (1) and ANN (2). Moreover, the larger the input vector size, the higher the over-fitting risk. For these reasons, the adoption of the optimized input array needs to be justified by a significant improved performance.

All the three different ANNs have been separately tested on the prediction of each output parameter (i.e. q, K_r and K_t) comparing the numerical values to the experimental ones.

In case of q, the optimization of the input set and all the further sensitivity analysis (Paragraph 4.4), were carried out by employing a restricted part of the wave overtopping database including only values of $q \geq 10^{-6}$ m^3/(sm). This decision was made by following the indications by Van Gent et al. (2007), who suggested not to train the ANN on values of $q<10^{-6}$ m^3/(sm) since these tests would be affected by a poor reliability. However, a specific chapter of this work (Chapter 6) is dedicated to the assessment of the reliability of such "small" q values and to the analysis of the ANN performance against these data.

For the quantitative examination of the ANN predictions, in this work three error indexes have been employed: the root mean square error, *rmse*, the Willmott index, *WI* (Willmott, 1981*)* and the coefficient of determination, R^2, respectively defined in Eq.s 4.3, 4.4 and 4.5. The quantity "*X*" (specified in Eq. 4.6), represents the output parameter, while the subscript indexes "s" and "ANN" respectively refer to the experimental and the predicted quantity.

$$\text{rmse} = \frac{1}{500}\sum_{i=1}^{500}\left(\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(X_{s_j}\text{-}X_{ANN,j}\right)^2}\right)$$

Eq. 4.3

$$\text{WI}=\frac{1}{500}\sum_{i=1}^{500}\left(1\text{-}\frac{\Sigma_{j=1}^{N}\left(X_{s,j}\text{-}X_{ANN,j}\right)^2}{\Sigma_{j=1}^{N}\left[\left|X_{s,j}\text{-}\overline{X}_s\right|+\left|X_{ANN,j}\text{-}\overline{X}_s\right|\right]^2}\right)$$

Eq. 4.4

$$\text{R}^2=\frac{1}{500}\sum_{i=1}^{500}\left(1\text{-}\frac{\Sigma_{j=1}^{N}\left(X_{s,j}\text{-}X_{ANN,j}\right)^2}{\Sigma_{j=1}^{N}\left(X_{s,j}\text{-}\overline{X}_s\right)^2}\right)$$

Eq. 4.5

$$\begin{cases} X = \begin{matrix} \log q'=\log q/\sqrt{g\cdot H_{m,0,t}^3} \\ K_r \\ K_t \end{matrix} \end{cases} \quad and \quad \overline{X}_s=\frac{1}{N}\sum_{j=1}^{N}X_{s,j}$$

Eq. 4.6

The adoption of different indexes is introduced to take into account different aspects of the ANN performance and to compare the results of the different applications. The *rmse* shows the dispersion of the predicted values with respect to the measured ones, therefore the lower the *rmse*, the more accurate the prediction. Its numerical value depends on the order of magnitude of the associated output parameter. In fact, the *rmse* values associated to q result at least 2-orders of magnitude larger than the *rmse* for K_r and K_t (see Tab. 4.6), because the values of q are processed in logarithmic scale (see Eq. 4.6).

WI and R^2 are instead normalized indexes and therefore range between 0 and 1, being 1 the perfect correspondence. R^2 accounts for the distribution of the experimental values around the mean, while *WI* accounts also for the distribution of the prediction with respect to the same experimental mean. *WI* is thus a symmetry indicator. The logarithmic transformation does not affect the normalized values of *WI* and R^2.

The so-called “large errors”, shown in the last column of Tab. 4.6, refer to the percentage of tests (with respect to the total number of tests) for which the ANN has systematically (in more than the 50% of resamples) predicted a value of the output parameter which differs from the experimental value more than the 50%.

The numerical values of the error indexes are reported in Table 4.6; these values correspond to the average results obtained from 50 bootstrap resamples of the database, and the uncertainty associated to each index is the standard deviation. Table 4.6 is organized into three parts according to the different applications.

Table 4.6 - Comparison among the quantitative performance of the three ANNs considered and tested. ANN (1) is the original CLASH ANN; ANN (2) is the ANN developed by Zanuttigh et al. (2013) for the prediction of the wave reflection; the new “optimized” ANN is the one defined within this work and finally proposed. Average results obtained from 50 bootstrap resamples of the database.

Prediction of the wave overtopping discharge, $q \geq 10^{-6}$ $m^3/(sm)$				
	RMSE	*WI*	R^2	# large errors (%)
ANN(1)	0.051 ± 0.003	0.971 ± 0.004	0.89 ± 0.01	2.5%
ANN(2)	0.048 ± 0.009	0.97 ± 0.02	0.91 ± 0.05	2.2%
New ANN	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
Prediction of the wave reflection coefficient, K_r				
	RMSE	*WI*	R^2	
ANN(1)	0.050 ± 0.009	0.985 ± 0.007	0.94 ± 0.03	7.8%
ANN(2)	0.039 ± 0.008	0.991 ± 0.007	0.96 ± 0.03	4.2%
New ANN	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
Prediction of the wave transmission coefficient, K_t				
	RMSE	*WI*	R^2	
ANN(1)	0.035 ± 0.007	0.9943 ± 0.004	0.97 ± 0.01	8.2%
ANN(2)	0.029 ± 0.007	0.996 ± 0.003	0.98 ± 0.01	6.7%
New ANN	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%

By comparing the results of the ANNs (Tab. 4.6), it is evident that ANN (1) performs its best when predicting q and K_t, and similarly ANN (2) when predicting K_r, consistently with the fact that these ANNs were originally optimized to predict these processes. ANN (2) yields to a significant reduction of the error in terms of error indexes with respect to ANN (1), confirming that the adoption of non-dimensional input elements and the introduction of the parameter D are key aspects for the improvement of the ANN performance.

Nevertheless, the new ANN returns the best outcomes for the application tests of overtopping and reflection, even in comparison with the correspondent specifically trained ANN (1) and ANN (2). This is probably due to the physically based scaling (with wave height and wave length) and the optimized input set, which includes some parameters (such as the toe width B_t) neglected by ANN (2).

For the wave transmission, the new ANN essentially provides the same results of ANN (2), which are indeed extremely satisfactory (*WI* is approximately equal to 1).

Therefore, the input set of the new ANN (Tab. 4.4) was selected as the most suitable, and hereinafter it is going to definitely characterize the new ANN and represent the reference one.

4.4 Architecture of the new ANN

The realization of an ANN model does not only comprehend the definition of the input parameters by means of a sensitivity analysis. In fact, the "*black-box*" nature of this kind of mathematical models requires a calibration process oriented to the choice and the optimization of several characteristic elements.

An ANN model essentially consists of layers (see Fig. 3.6). The input elements are "connected" to one or more hidden layers, each of them being composed by an undefined number of hidden neurons. The ANN elaborates the numerical information contained in the input and passes it to the first (or unique) hidden layer through the so-called "hidden layer transfer function", a proper mathematic function that transforms the numerical values of the input neurons. Similarly, the following hidden layers – if existing – receive the information from the hidden layer through other transfer functions, up to the output layer (in this case, we have the "output neuron transfer function").

Each input element is connected to each hidden neuron, and the hidden neurons are likewise connected to the output. The process of training the ANN essentially consists in the definition of the numerical weights to be assigned to each connection: this process, governed by the training algorithm, is based on the iterative research of the minimum error among the outputs predicted by the ANN itself and the experimental target values. The velocity and the efficiency of the training depend on the transfer functions, the error type to minimize, the tolerance imposed and on the training algorithm (more details about these issues can be found in Chapter 2).

The architecture of the optimized ANN has been directly derived from ANN(2), based on the analysis presented already by Zanuttigh et al. (2013) and Formentin et al. (2013), and then tested against the prediction of all the three processes.

The resulting optimal characteristics of the new ANN architecture are conceptually portrayed in Fig. 4.5 and resumed in the following :

- multi-layer network, based on a "feed-forward back-propagation" learning algorithm; 1 hidden layer, and 1 output neuron, corresponding either to K_r, K_t or q;
- the hidden layer comprehends 40 hidden neurons; this number has been re-defined after a specific sensitivity analysis (see the reserved subsection later in this paper);
- training algorithm: Levenberg – Marquardt (Levenberg, 1994; Marquardt, 1963);
- hidden neurons transfer function: hyperbolic tangent sigmoid function;
- output neuron transfer function: linear transfer function.
- error type: mse (mean squared error);
- maximum number of iterations (epochs) allowed: 100;
- method to improve generalization: none, after testing (and deciding to discard) the "early stopping" method. The assessment of the ANN performance and its capability of generalization are attributed to the bootstrap resampling technique.
- adoption of the weight factors (WF) to drive the bootstrap resampling: no.

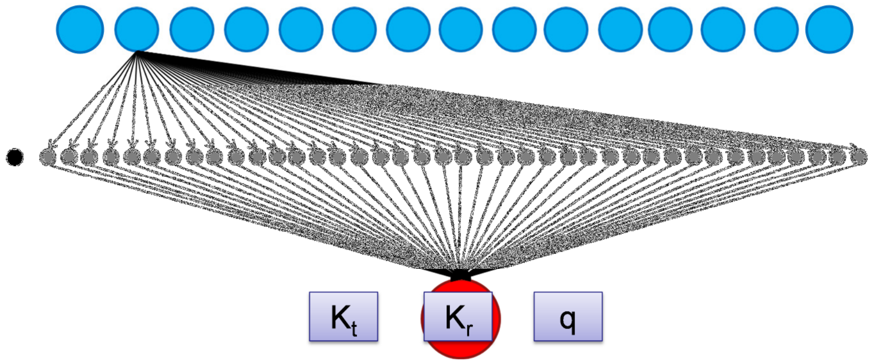


Figure 4.5 – Schematization of the conceptual layout of an ANN, organized in layers. The number of the input elements (15), the hidden neurons (40, and 1 bias) and the output neuron (1, alternatively *q*, *K$_r$* and *K$_t$*) reflects the architecture of the network definitely chosen in this work, i.e. the new "optimized" ANN.

Although each of these parameters was defined after calibration, only the most significant analysis are reported within this paper. In particular, the first subsection in the following, describes in detail the optimization of the number of hidden neurons, while the second one focuses on the different techniques to improve the capability of generalization of the ANN.

4.4.1 The bootstrap resampling technique

The bootstrap technique consists in several (N) resampling with replacement of the data to be selected for the training of the ANN. For each run of the ANN, one of the N bootstrapped databases is used for the training. The size of each bootstrapped database equals the original one, but the included data are differently assorted, since each selected test is randomly selected with replacement. Each bootstrapped database for each training of the ANN may therefore include the same tests more than once, while some tests may never appear.

The bootstrap resampling of the database is principally adopted to assess the performance of an ANN. Each differently-trained ANN yields to differently evaluated output parameters, and the ensemble of the predicted outputs can be considered as a stochastic variable and therefore used to derive average indexes of performance and standard deviations.

Furthermore, if the number of resamples is large-enough to be statistically significant, it is possible to calculate the quantiles of the distribution and derive the confidence intervals. A paramount aspect is that a mean prediction is not only more significant from a statistical view-point, but is also more accurate since it adopt the commitment of several randomly-trained ANNs.

Tests have been carried out by means of different numbers of bootstrap resamples, ranging from 1 up to 500, based on the suggestions from previous works (Van Gent et al., 2007; Verhaeghe et al., 2008).

The results of these tests – carried out for each application case – do not show any significant dependence on the increase of the number of resamples above 50, as it can be appreciated by observing the values of the error indexes reported in Tab. 4.7 and the diagrams of Fig. 4.6.

In details, the diagrams of Fig. 4.6 respectively reports the trend of the average *rmse* (plot to the left) and its corresponding values of standard deviation (plot to the right) for the application case of the wave reflection, selected as example. From these plots, it can be observed that the average value of *rmse* stabilizes around an almost constant value of 0.038 for a number of bootstrapping resamples greater than 30, while its standard deviation continues to monotonically decrease till reaching the value of 0.004 for a number of resamples greater than 80. Considering that between 30 and 80 simulations the value of the standard deviation decreases just from 0.006 to 0.004, it has been decided to choose the intermediate number of 50 bootstrap resamples as the optimal to characterize the network performances.

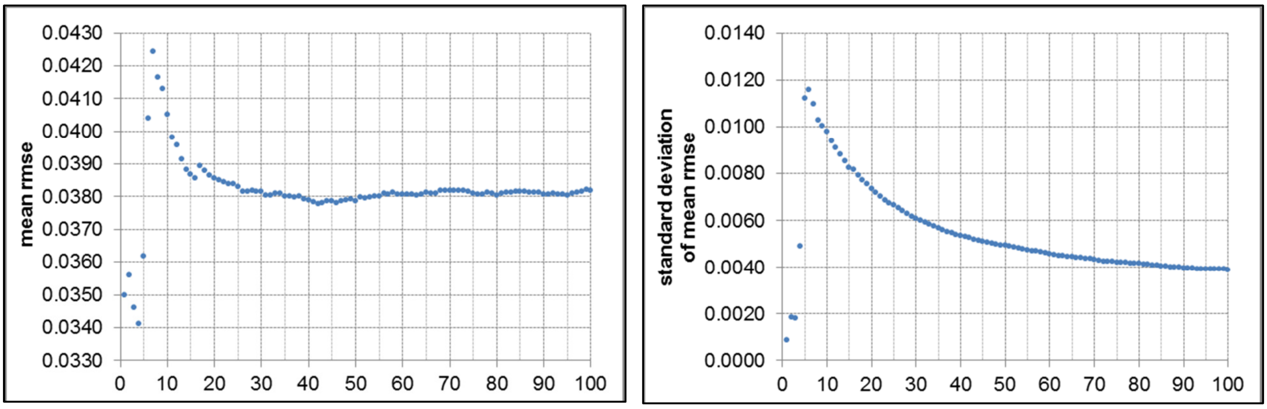


Figure 4.6 – Mean *rmse* value (left, ordinate) and associated standard deviation value (right, ordinate), as functions of the number of bootstrapping resamples (left and right, abscissa). Results produced by the "optimized" ANN tested against the prediction of the prediction of the wave reflection coefficient K_r.

Tab. 4.7 provides the comparison among the average ANN results obtained after 50 and after 500 bootstrapping resamples. These results essentially prove that the difference is made by the use or not of the resampling technique (see also Paragraph 4.4.3), and not by the adoption of an extremely high number of resamples. The number of 50 resamples is confirmed to be widely sufficient, and represents a good compromise to speed up the sensitivity analysis to other parameters. This can be justified because of the very large size of the databases.

Table 4.7 – Comparison among the quantitative performance of the new ANN trained on 50 and 500 bootstrapping resamples of the databases.

New ANN - Prediction of the wave overtopping discharge, $q \geq 10^{-6}$ m^3/(sm)				
Nr of. resamples	*RMSE*	*WI*	R^2	# large errors (%)
50	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
500	0.046 ± 0.008	0.98 ± 0.01	0.91 ± 0.05	1.8%
New ANN - Prediction of the wave reflection coefficient, K_r				
Nr of. resamples	*RMSE*	*WI*	R^2	
50	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
500	0.04 ± 0.01	0.99 ± 0.01	0.96 ± 0.04	4.2%
New ANN - Prediction of the wave transmission coefficient, K_t				
Nr of. resamples	*RMSE*	*WI*	R^2	
50	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%
500	0.03 ± 0.01	0.996 ± 0.009	0.98 ± 0.04	6.0%

4.4.2 The number of hidden neurons

The definition of the number of hidden neurons – which represents one of the key-features of an ANN – is generally related to the number of input parameters and to the range of variability of the input data, but it cannot be defined *a priori*.

The common methodology (Van Gent, 2007; Panizzo and Briganti, 2007; Verhaeghe, 2005) to establish the optimal number of hidden neurons by testing the performance of the optimized ANN as a function of the progressive increase of the number of the hidden neurons has been here adopted. Since the ANN is supposed to predict K_r, K_t and q, all the applications have been taken into account, and the optimal number has been finally established as the most suitable for all the outputs.

The results of the sensitivity analysis are graphically reported in Figure 4.7 in terms of average *rmse* values and standard deviation. These values were derived from the resulting prediction of the optimized ANN after 50 bootstrap resampling of the training database.

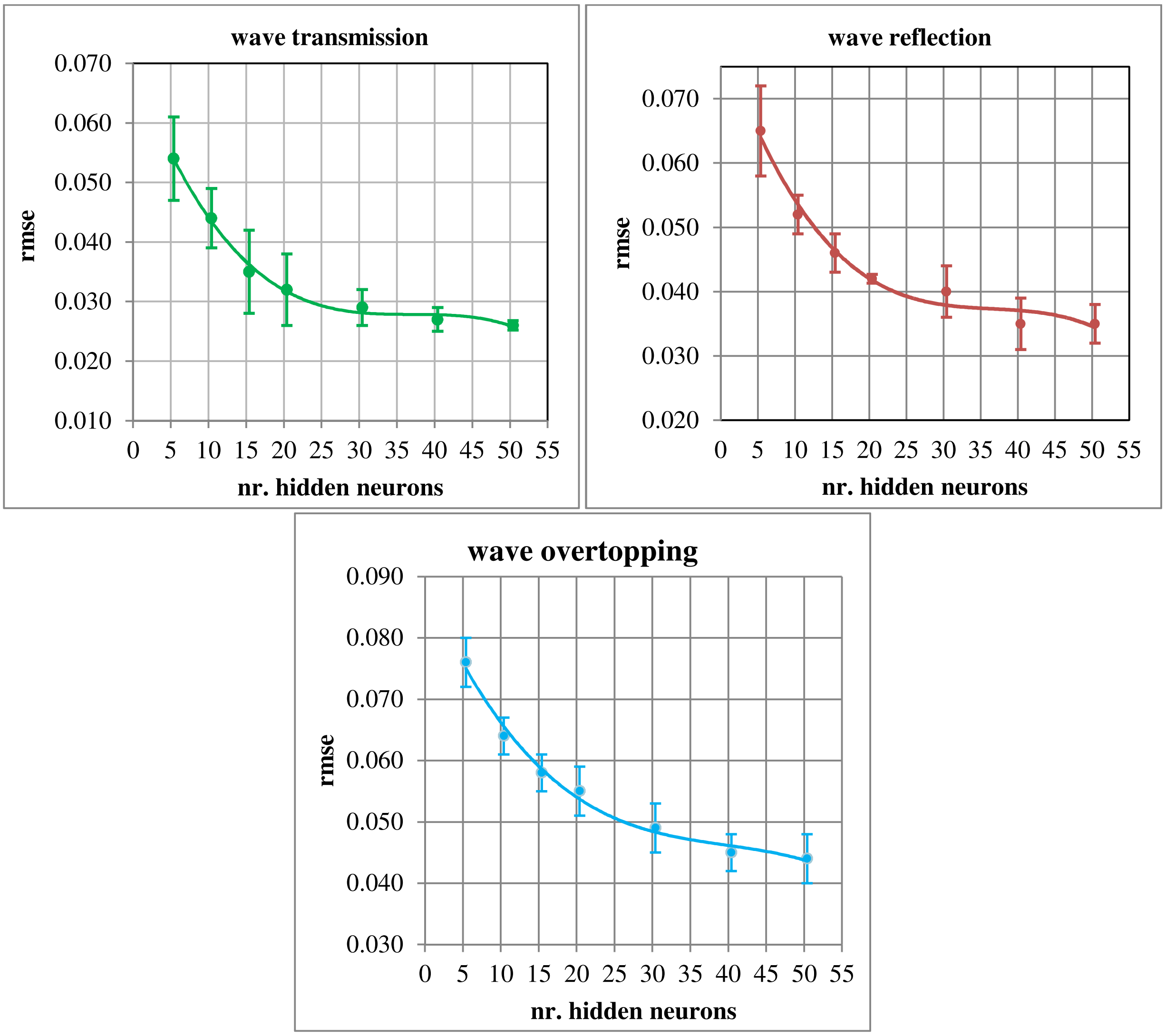


Figure 4.7 - Mean *rmse* values (ordinate) and corresponding standard deviations as functions of the number of hidden neurons (abscissa); from left to right and from top to bottom: wave transmission, reflection and overtopping. Results performed by the new ANN and averaged after 50 bootstrap resamples.

The computation of average errors allows to detect (Fig. 4.7) how the increase of the number of hidden neurons may induce not only a reduction of the *rmse* but also a decrease of the error band, i.e. of the standard deviation. The analysis of this latter aspect, which was not considered by previous works, shows that the ANN tends to systematically perform better and therefore to be less affected by the random selection of data for training by increasing the size of the hidden layer. An important consequence is the reduction of the uncertainty associated to the prediction, which corresponds to narrower intervals of confidence for each output estimation.

The trend of the number of hidden neurons in predicting *q* (Fig. 4.7) would suggest to select 50 hidden neurons, both considering the *rmse* value and its standard deviation. However, the predictions of K_t and K_r show that the number of 50 hidden neurons would not represent a good compromise among the complexity of the architecture and the performance obtained, since the reduction of *rmse* is negligible (K_t) or even opposite (K_r) when the number of hidden neurons is greater than 40. Differently, the trend of *q* still shows a little improvement of performance at the increase of hidden neurons from 40 to 50, and this can be explained by the sensibly greater number of tests available for the wave overtopping database (see Paragraph 4.2). Nevertheless, the polynomial interpolating curve presents a flection point after 40 hidden neurons for all the applications, revealing the beginning of the over-fitting.

The complexity of the architecture is indeed a paramount issue, since from one hand it characterizes the minimum number of tests to training the ANN (see Tab. 4.5), and from the other hand it may induce overtraining problems (see the following Paragraph 4.4.3 regarding the improvement of the generalization). Taking into account all these correlated aspects, a final number of 40 hidden neurons was selected.

4.4.3 Improving generalization: the “early-stopping” technique

One of the most important issues correlated to the actual performance of an ANN is represented by its capability of generalization, i.e. of overcoming the limits of the range of training tests (see Paragraph 2.2.2). An ANN is said to be “over-trained” when it is able to reproduce very well its training data, but is not able to predict with sufficient accuracy beyond the training ranges. This generally occurs when a “too-large” number of hidden neurons has been applied and the architecture of the ANN is “too much focused” on replicate the trend of the training tests and does not learn the “general rule”.

Since a pretty-large number of hidden neurons is anyway requested to obtain satisfactory results (see Fig. 4.8), especially if a wide database is employed, several techniques to improve generalization are available. The most common technique is the so-called “early-stopping”: this methodology essentially consists in splitting the overall database into three datasets, training the ANN only over one of them (the proper “training set”) and using the remaining sets to stop if needed the training process before the achievement of the expected performance. The stopping is imposed when for several iterations consecutively the *rmse* on the validating set is not reduced, even if the training *rmse* continues to decay. In other terms, the “early-stopping” technique interrupts the training process before the “optimum”, leading to a slightly worse performance, but ensuring a greater capability of generalization.

Therefore, the validating set is employed to test the ANN capability of generalization, i.e. the capability to predict values not belonging to the training set, while the testing set is employed to verify the performance of the ANN at each iteration without affecting the training process.

The adoption of the bootstrap resampling of the database allows assessing the capability of generalization of an ANN avoiding the implementation of specific methodologies, such as the early stopping (Verhaeghe, 2005). In fact, an ANN trained many times, each time on a randomly different (bootstrapped) database, produces “average” predictions and relative indexes of performance, such as the standard deviation or intervals of confidence (see the related section below).

The effects of the implementation of the "early-stopping" were considered, through a specific sensitivity analysis. Table 4.8 synthetizes the performance of the ANN with respect to all the applications (q, K_r and K_t), comparing the results obtained with the "early-stopping" and without it. Similarly to the sensitivity analysis to the input parameters (see Tab. 4.4), the error indexes *rmse*, *WI* and R^2 have been calculated. The numerical values were derived as average results from 50 different random splitting of the database into training-testing-validating sets (in case of early stopping) and from 50 bootstrap resampling of the database.

The introduction of the "early-stopping" causes a sharp reduction of the ANN performance, as it can be appreciated by the increase of the *rmse*, and the contemporary decrease of *WI* and R^2 (Table 4.8). Therefore, the choice to exclude the early stopping and to assess the uncertainty of the predictions with the bootstrapping technique is straightforward.

Table 4.8 – Synthesis of the performance of the new ANN obtained with and without the "early-stopping" technique. Average results from 50 bootstrap resamples of the database (case without early stopping) and 50 different random splitting of the database into training-testing-validating sets (case with early stopping).

New ANN - Prediction of the wave overtopping discharge, $q \geq 10^{-6}$ m^3/(sm)				
Early stopping	*RMSE*	*WI*	R^2	# large errors (%)
No	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
Yes	0.05 ± 0.07	0.97 ± 0.07	0.9 ± 0.6	3.1%
New ANN - Prediction of the wave reflection coefficient, K_r				
Early stopping	*RMSE*	*WI*	R^2	
No	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
Yes	0.05 ± 0.02	0.99 ± 0.02	0.9 ± 0.1	7.4%
New ANN - Prediction of the wave transmission coefficient, K_t				
Early stopping	*RMSE*	*WI*	R^2	
No	0.029 ± 0.009	0.996± 0.005	0.98 ± 0.02	8.4%
Yes	0.036 ± 0.009	0.989 ± 0.006	0.98 ± 0.03	12.7%

4.4.4 The weight factors

The methodology adopted by Van Gent et al. (2007) to perform the bootstrap resampling suggests the employment of the weight factors *WF*, functions of the two general parameters *RF* and *CF* (see Tab. 4.1). The employment of the *WF* implies "driving" the random selection of the data for the training according to the weight attributed to each test. The higher the *WF*, the higher the probability for a test to be selected.

However, the attribution of the parameters *RF* and *CF* to each test may be affected by a certain subjectivity, according to the laboratory or the team who performed the tests, especially for the definition of *RF*. Moreover, the low weighting of some very complex tests (reporting *CF*=3) may cause an opposite effect of worsening the overall ANN performance. Indeed, the composite the cross-section of a structure, the increased necessity for the ANN to recognize that pattern and therefore the increased necessity to select that test during the training phase.

For this reason, the new ANN was trained considering both the weighting of the data with the *WF* and the exclusion of them from the training phase. The final advisable methodology to train the new ANN resulted in the exclusion of the *WF* from the bootstrapping, as the numerical results reported in Tab. 4.8 indicate. For each application, the introduction of the *WF* generates an increase of either the number of large errors (q and K_r) and of the standard deviations (q and K_t). Besides, for K_r a clear worsening of the performance is detected when applying the *WF*, considering both the error indexes and the number of large errors, which nearly duplicates.

Table 4.9 – Synthesis of the performance of the new ANN obtained including or excluding the weight factors (WF) from the bootstrap resampling of the database. Average results from 50 resamples of the database.

New ANN - Prediction of the wave overtopping discharge, $q \geq 10^{-6}$ m^3/(sm)				
WF	*RMSE*	*WI*	R^2	# large errors (%)
No	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
Yes	0.048 ± 0.005	0.974 ± 0.008	0.91 ± 0.03	2.5%
New ANN - Prediction of the wave reflection coefficient, K_r				
WF	*RMSE*	*WI*	R^2	
No	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
Yes	0.048 ± 0.009	0.987 ± 0.008	0.95 ± 0.03	7.4%
New ANN - Prediction of the wave transmission coefficient, K_t				
WF	*RMSE*	*WI*	R^2	
No	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%
Yes	0.03 ± 0.01	0.996 ± 0.007	0.98 ± 0.03	7.1%

Please, note that the "exclusion" of the *WF* exclusively regards the process of bootstrapping. This means that the data reporting *RF* or *CF* = 4 were *a priori* discarded from the training process since – following the suggestions by Van Gent et al. (2007) – these tests were considered unreliable or too much complex for neural network modelling. Actually, it was verified that including these tests generates increased scatter. The results of this further sensitivity analysis are not reported here.

4.5 Preliminary steps that leaded to the definition of the new ANN

This section aims to illustrate all the main kinds of analysis, selection processes and techniques followed to optimize the performance of the ANN and define its layout. The new ANN presented in the Paragraphs 4.3 and 4.4 was essentially derived from a previous network primarily developed for the representation of the wave reflection, and in a second moment applied to the wave transmission. The layout of this preliminary ANN was partially different, and it was gradually optimized by means of a revision and extension of the database and of comparisons with existing ANNs and formulae. The analysis and the results presented in the following, based on the preliminary ANN, definitely provide an idea of these steps.

The paragraph is then subdivided into two Paragraphs 4.5.1 and 4.5.2, respectively focused on the preliminary ANN for the wave reflection and the wave transmission. Both these Paragraphs are in turn organized into subsections that present the specific analysis carried out: the evaluation of the ANN performance and the study of the error distribution, the calibration and the selection of some input parameters, the testing of the ANN to the extension of the database beyond the range of training, the validation of the ANN against the existing formulae.

4.5.1 Preliminary ANN for the wave reflection

The first step followed to evaluate the performance of an ANN is the analysis of the qualitative and quantitative comparison among measured and predicted values of the output (in this case, K_r).

The qualitative performance is provided by plots such as the ones reported in Figure 4.8: the diagram to the left shows the computed values of K_r, i.e. $K_{r,ANN}$, are compared with the measured values, $K_{r,s}$. The central line represents the bisector, i.e. the perfect correspondence among predicted and experimental values, while the external lines represent the 95% confidence boundaries. The diagram to the right displays the dispersion of the absolute error e = $K_{r,s}$ - $K_{r,ANN}$ as a function of $K_{r,s}$.

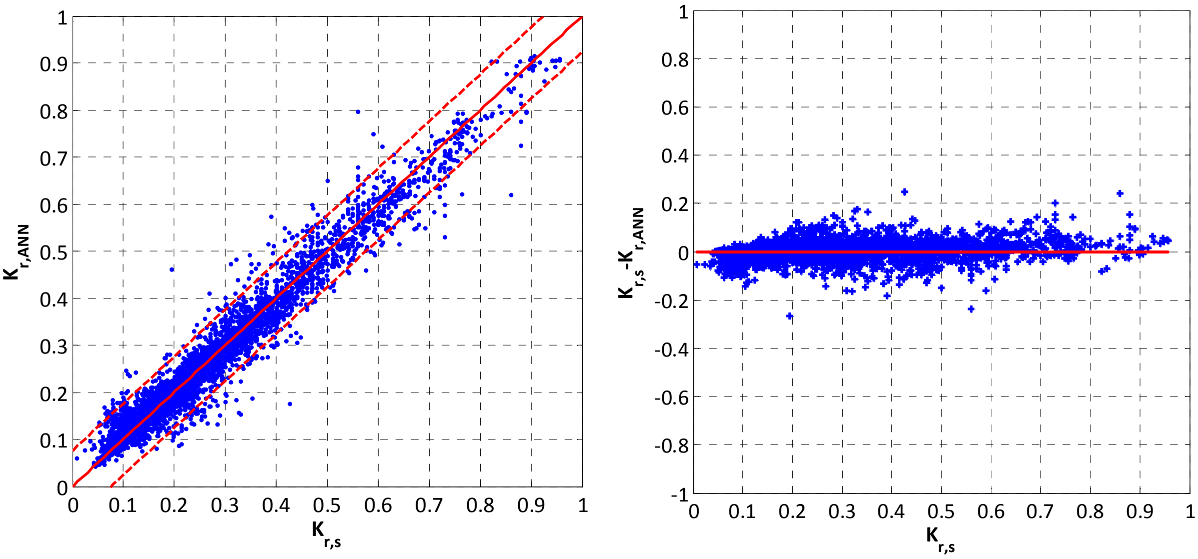


Figure 4.8 – Left: Comparison among K_r predicted values ($K_{r,ANN}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); the continuous bisector represents the ideal condition ($K_{r,ANN}$ = $K_{r,s}$), while the dashed lines refer to the 95% confidence levels. Right: difference e = $K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of $K_{r,s}$ (abscissa). Preliminary ANN.

In this case, both the graphs of Fig. 4.9 reveal a pretty good agreement of computations and measurements; the confidence interval is quite narrow and, moreover, the ANN appears to provide the results with a good degree of symmetry, as it can be appreciated also by the histogram in Figure 4.9.

The quantitative estimate of the ANN accuracy is provided through the same above-mentioned (Paragraph 4.3) error indexes *rmse*, *WI* and R^2 (for the definitions, see Eq.s from 4.3 to 4.6) and the computation of the number of recurrent "large" errors. Tab. 4.10 reports the values of these indexes associated to the preliminary ANN for K_r. It is important to remark that for this preliminary ANN, the technique of the "early-stopping" is applied and the bootstrap resampling of the database not yet implemented. Therefore, the results displayed in Tab. 4.8 correspond to the average values obtained from 50 different random splitting of the database into training-testing-validating sets and 50 consequent simulation of the ANN.

Table 4.10 - Synthesis of the performance of a preliminary ANN optimized for the prediction of the wave reflection coefficient; average results from 50 different random splitting of the database into training-testing-validating sets.

Preliminary ANN optimized for the prediction of the wave reflection coefficient, K_r			
RMSE	*WI*	R^2	# large errors (%)
0.038 ± 0.003	0.985 ± 0.003	0.943 ± 0.006	2.5%

In Tab. 4.10, the *rmse* value of 0.038 is particularly good, since it essentially corresponds to the one of the "optimized" new ANN and its standard deviation of 0.003 is sensibly lower (see Tab. 4.6). This result is just as good if compared to the existing formulae for the prediction of the wave reflection (see Paragraph 4.5.1.4), characterized by larger values of *rmse* and associated to well defined structures typologies and therefore more restricted datasets (Zanuttigh and van der Meer, 2008; Zanuttigh and Lykke Andersen, 2010).

The high *WI* value of 0.985 denotes that the error distribution is not just narrow around the optimal condition, but that it is also satisfactory symmetric. The values of the standard deviation of *rmse, WI* and R^2 are all approximately 10^{-3}, denoting the ANN's stability against the random data-selection processes (training and testing). The model is therefore consistently accurate and reliable.

However, the comparison with the new ANN, which provides higher values of *WI* and R^2 (respectively, 0.992 and 0.97, see Tab. 4.6) enhances that this preliminary ANN was not completely optimized (especially considering the matter of the bootstrap resampling).

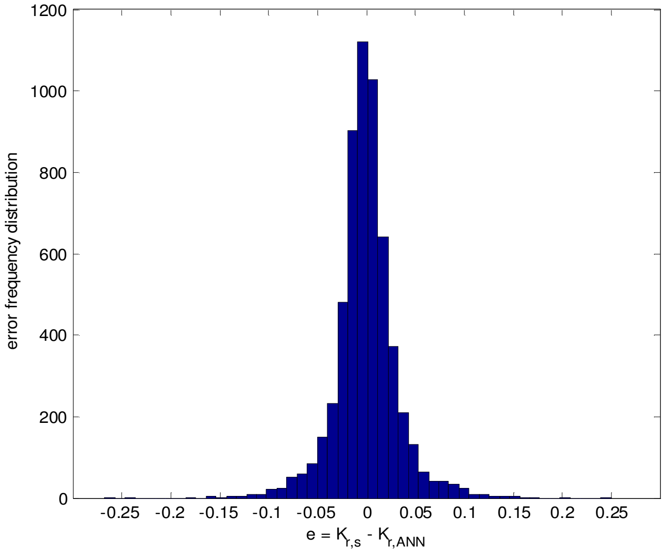


Figure 4.9 – Difference $e = K_{r,s}$ - $K_{r,ANN}$ frequency distribution histogram

Besides the examination of the error indexes, a greater detail of comprehension of the performance of an ANN can be achieved by analyzing the tests affected by the larger and more recurrent errors.

For this purpose, within the previous work focused on the preliminary ANN for the wave reflection, an in-depth analysis of the error was carried out, through the definition of *ad-hoc* indexes and threshold values to characterize the error distribution.

Differently from the definition of the "large" errors provided in Paragraph 4.3, during the previous work a "threshold" value of $e = K_{r,s}$ - $K_{r,ANN}$ = |0.15| was set. Being the average value of $K_{r,s} \approx 0.26$ (see Tab. 4.11), the value of 0.15 represented, on an average, a percentage relative error of approximately 50%. Errors lower than |0.15| were supposed to fall within the random uncertainty, while greater errors deserve a more accurate analysis.

Subsequently, the errors obtained from the 50 simulations of the ANN were analyzed. Both the diagrams of Figure 4.10 aim to show all the data for which the ANN computed an absolute error $e \geq |0.15|$ at least once during the 50 simulations. The panel to the lefts report the errors greater than |0.15| as a function of the test indexes, while the panel to the right adopts the experimental $K_{r,s}$ values as abscissa. From these plots, the data repeatedly affected by high errors are immediately visible, thanks to the concentration of error points aligned on its corresponding indices (to the left) or values of $K_{r,s}$ (to the right).

If the same test is systematically affected by a high error, it may imply either that the test itself is less reliable (for example, due to measurement errors especially in case of low $K_{r,s}$ or inaccurate methodology of analysis) or that the ANN does not correctly learn the input-output relation for the test. Based on the diagram to the left of Figure 4.10, it is clear that large errors occur especially for specific values of the indexes, which correspond to structures with berms or seawalls (indexes values >4500) and smooth impermeable structures (3000÷3500).

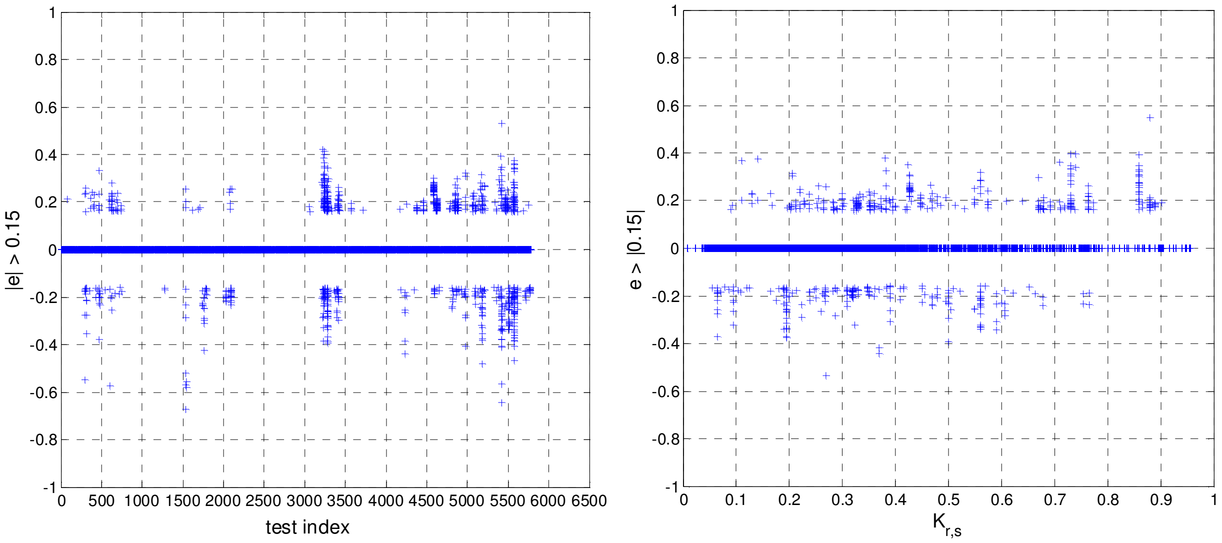


Figure 4.10 – Left: |*e*| = |$K_{r,s}$-$K_{r,ANN}$ | ≥ *0.15* (ordinate) as a function of database test indexes (abscissa); right: |*e*| = |$K_{r,s}$-$K_{r,ANN}$ | ≥ *0.15* (ordinate) as a function of $K_{r,s}$ (abscissa).

Differently, based on the plot to the right of Figure 4.10, there is no evident specific concentration of large errors depending on the values of $K_{r,s}$. The preliminary ANN however tends to overestimate the very low values of $K_{r,s}$ ($K_{r,s} \ll 0.1$) and to underestimate the very high values of $K_{r,s}$ ($K_{r,s} > 0.85$), showing the typical ANN problems occurring with extrapolation.

From a quantitative viewpoint, the preliminary ANN made 508 errors greater than |0.15| over the 50 simulations (i.e., altogether over more than 280'000 data), i.e. a frequency of 0.2% of "large" errors (e_{large}) was estimated.

In order to quantify the average "large" error provided by the preliminary ANN and compare it to the global ANN performance, two relative error indicators were defined and the definitions are here reported through Eq.s. 4.7 and 4.8. The results – synthesized in Table 4.11 – show that the preliminary ANN would have produced errors larger than 80% ($\bar{e}_{large,\%} = 83\%$) just "twice over 1000 data" (0.2%), while on average the ANN produced errors approximately lower than 10% ($\bar{e}_{\%}$=8.3%).

$$\bar{e}_{\%}=\frac{\bar{e}}{\bar{K}_{r,s}}\cdot 100,\ \ where\ \bar{e}=\frac{1}{50}\cdot\sum_{i=1}^{50}\left(\frac{1}{5781}\cdot\sum_{j=1}^{5781}|e_{j,i}|\right),$$

Eq. 4.7

$$\bar{e}_{large,\%}=\frac{\bar{e}_{large}}{\bar{K}_{r,s}}\cdot 100,\ \ where\ \bar{e}_{large}=\frac{1}{508}\cdot\sum_{k=1}^{508}\left|e_{large}\right|,$$

Eq. 4.8

This analysis allowed detecting some "unreliable" tests and some mistakes present in the wave reflection database due to copy errors presumably committed during the process of gathering and homogenization of the tests from the different datasets to the complete database. Furthermore, the idea to identify the higher and most recurrent errors has leaded to the definition of the "large" errors (see Paragraph 4.3) employed for the assessment and characterization of the new ANN performance.

Table 4.11 – Values of the average absolute and relative errors provided by the preliminary ANN for the wave reflection.

Preliminary ANN for the wave reflection				
$\bar{e}$	$\bar{e}_{\%}$	$\bar{e}_{large}$	$\bar{e}_{large,\%}$	$\bar{K}_{r,s}$
0.021	8.3%	0.212	83%	0.255

4.5.1.1 Representation of structures with non-straight slopes

A specific investigation for the "optimal" input parameters for the representation of structures with berms and non-straight slopes was carried out for the preliminary ANN. The results of this sensitivity analysis are particularly important, since the "optimized" input parameters found out within this previous work are the same adopted in the final layout of the new ANN (see Tab. 4.4).

To simulate the structures with berms, toe protections and foreshores, it is necessary to introduce some other specific parameters, such as $G_c/L_{m-1,0,t}$, $B/L_{m-1,0,t}$, $h_b/H_{m,0,t}$, cotα_{incl} and m. The calibration of these parameters was, at first, carried out by training the preliminary network on the datasets E and F (see Fig. 4.3 and Tab. 4.3) instead of the complete database.

As example, just one of all the sensitivity tests is here reported, i.e. the process which leaded to the choice of $h_b/H_{m,0,t}$ instead of $h_b/L_{m-1,0,t}$.

The ratio $h_b/H_{m,0,t}$ revealed to be more suitable and experimentally satisfactory than $h_b/L_{m-1,0,t}$ to reproduce the berm submergence (i.e. wave breaking on the berm). Indeed, $h_b/H_{m,0,t}$, besides providing a better performance, was more sensitive to the increase of the number of hidden neurons. When adopting $h_b/H_{m,0,t}$, *rmse* decreases of 3% and the 95% confidence intervals are therefore narrower, as detectable by comparing the diagram to the left of Figure 4.11 (which exactly corresponds to Fig. 4.8, left) with the one to the right of the same Figure 4.11. Moreover, the ANN with $h_b/L_{m-1,0,t}$ tends to under-estimate the actual values of K_r, as it is shown by the larger number of scattered points under the continuous bisector line in Figure 4.11, right.

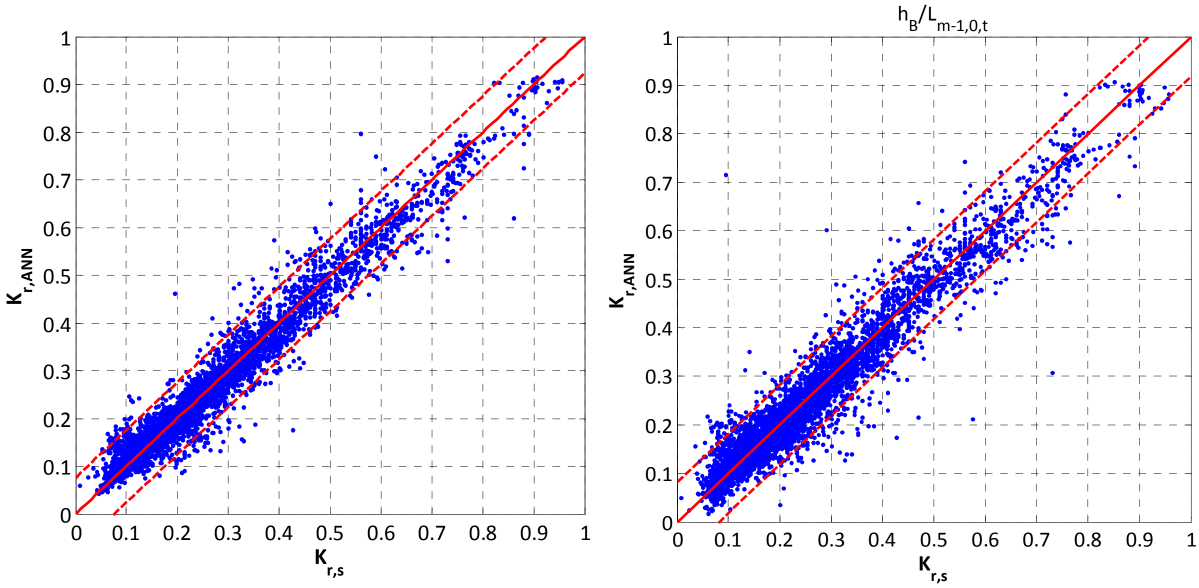


Figure 4.11 – Comparison among K_r predicted values ($K_{r,ANN}$, ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); to the left: preliminary trained with $h_b/H_{m,0,t}$; to the right: preliminary ANN trained with $h_b/L_{m\text{-}1,0,t}$.

Finally, the sensitivity of both parameters to the hidden neurons is shown in Figures 4.12 and 4.13: they respectively represent the values of *rmse* and *WI* as functions of the increasing number of hidden neurons, respectively. Each of the values refers to the average of the 50 simulations performed.

The values of the standard deviation, as the values of *rmse* themselves, are generally larger for the parameter $h_b/L_{m\text{-}1,0,t}$. The better performance of the preliminary ANN with $h_b/H_{m,0,t}$ can be also appreciated by comparing the best fitting curves of *rmse* and *WI* in Figures 4.12 and 4.13, where the curves for $h_b/L_{m\text{-}1,0,t}$ are above and below the curves fitting $h_b/H_{m,0,t}$, respectively. Besides, the element $h_b/H_{m,0,t}$ clearly shows a greater sensitivity to the increasing dimension of the hidden layer: the ANN over-fitting is evident for a number of hidden neurons greater than 25, since the fitting curves begin to oscillate.

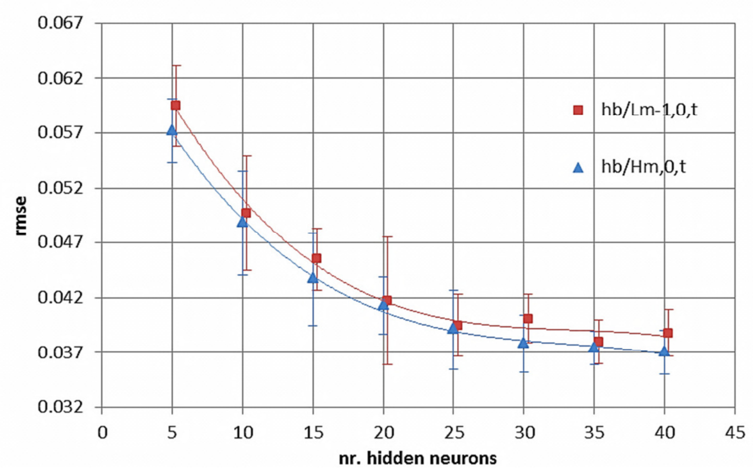


Figure 4.12 – *rmse* (ordinate) as a function of the number of hidden neurons (abscissa); the square points are associated to the preliminary ANN trained with $h_b/L_{m-1,0,t}$ and the triangle points to the network with $h_b/H_{m,0,t}$. Average results obtained from 50 simulations of the ANN.

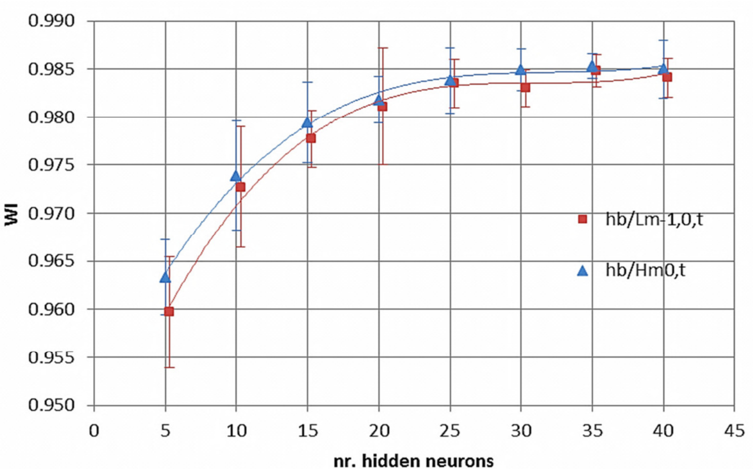


Figure 4.13 - *WI* (ordinate) as a function of the number of hidden neurons (abscissa); the square points are associated to the preliminary ANN trained with $h_b/L_{m-1,0,t}$ and the triangle points to the network with $h_b/H_{m,0,t}$. Average results obtained from 50 simulations of the ANN.

4.5.1.2 Representation of oblique wave attacks conditions

The modelling of structures under oblique wave attacks represents an interesting test bed to analyze an ANN capability of generalization. For oblique cases, i.e. the dataset G, the relative crest freeboard $R_c/H_{m,0,t}$ varies within the range [-0.66; 1.90], while in the full reflection database the ranges of values of $R_c/H_{m,0,t}$ are [-8.10; 25.40]. It is therefore important to understand if the network can well-estimate K_r in oblique conditions for the whole range of $R_c/H_{m,0,t}$.

The preliminary reflection ANN was tested against an extension of the database. Indeed, an artificial set of data characterized by a fixed value of $\beta \neq 0$ and out-of-range values of the relative submergence $R_c/H_{m,0,t}$ was created. The artificial data were built up so that all the input elements maintained constant values (summarized in Table 4.12), with the exception of $R_c/H_{m,0,t}$, whose range spaced between [-2.00; +10.00]. Since the whole training set was constant, K_r simply became a function of $R_c/H_{m,0,t}$, and therefore it was expected to increase at the increasing of $R_c/H_{m,0,t}$. In other terms, it was possible to predict the values of K_r and notice if the network behaved as expected or not.

Table 4.12 – Constant values of the input elements artificially built up to test the preliminary ANN sensitivity to $R_c/H_{m0,t}$ in oblique wave attack cases.

Preliminary ANN for the wave reflection	
Input element	value
$H_{m,0,t}/L_{m-1,0,t}$	0.036
$h_t/L_{m-1,0,t}$	0.170
γ_f	0.55
$cot\alpha_d$	0.962
$cot\alpha_{incl}$	2.000
$D/H_{m,0,t}$	0.385
$R_c/H_{m,0,t}$	[-2.00; 10.00]
$B/L_{m-1,0,t}$	0.00
$h_b/H_{m,0,t}$	0.00
$G_c/L_{m-1,0,t}$	0.046
m	1000
β	60.00
Spreading	5.00

Figure 4.14 shows the results of such sensitivity test. $K_{r,ANN}$ monotonically increases with $R_c/H_{m,0,t}$, but it is upper limited (i.e. $K_{r,ANN} < 1$), as it is remarked by the continuous fitting line. This allows concluding that the preliminary ANN was apparently able to predict K_r for oblique wave attacks even out of the range of training values of $R_c/H_{m,0,t}$. The complete training database was evidently wide enough to make the ANN providing correct predictions also in various conditions.

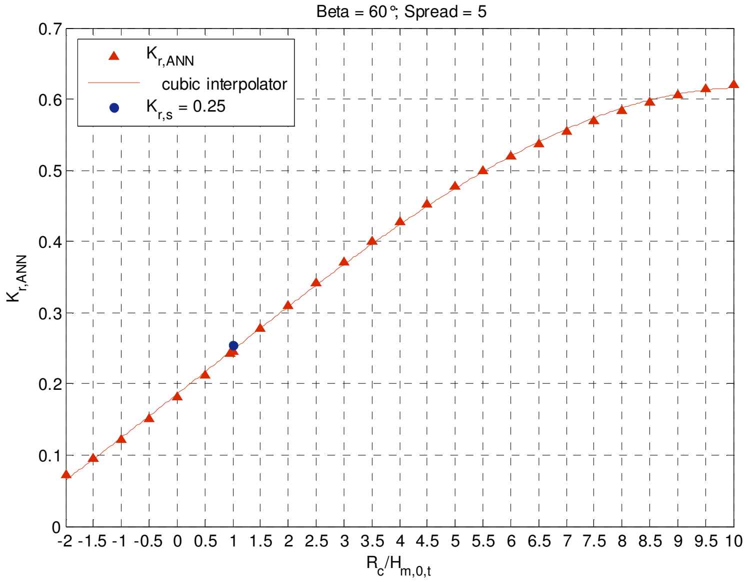


Figure 4.14 – Predicted $K_{r,ANN}$ values (ordinate, red triangles) as a function of $R_c/H_{m,0,t}$ (abscissa) during the test of sensitivity in oblique wave attacks. The blue circle represents the $K_{r,s}$ value of the original test, characterized by $R_c/H_{m,0,t} = 0.962$. The continuous line is a cubic interpolator of predicted data. Preliminary ANN.

4.5.1.3 Wave directional spreading

The input element representing the wave directional spreading was introduced in the preliminary ANN for the wave reflection but excluded from the final input set of the optimized ANN (see Tab. 4.4) since an *RF = 4* has been applied to all the tests corresponding to short-crested waves (i.e. waves presenting *spreading* ≠ 0).

The need of including an input element such as the wave directional spreading in the preliminary ANN was investigated thoroughly, by analyzing the predictions of the ANN in case of oblique and 3D wave attacks.

Taking into account the wave directional spreading allows to detect the different behaviour of short-crested and long-crested waves, therefore, it yielded to an improvement of the ANN performance (a decrease of *rmse* of 8%).

Figure 4.15 shows the preliminary ANN performance derived by eliminating and adding the spreading parameter. The comparison among the values of $K_{r,ANN}$ and $K_{r,s}$ is presented just for oblique wave attacks distinguishing among short and long-crested waves.

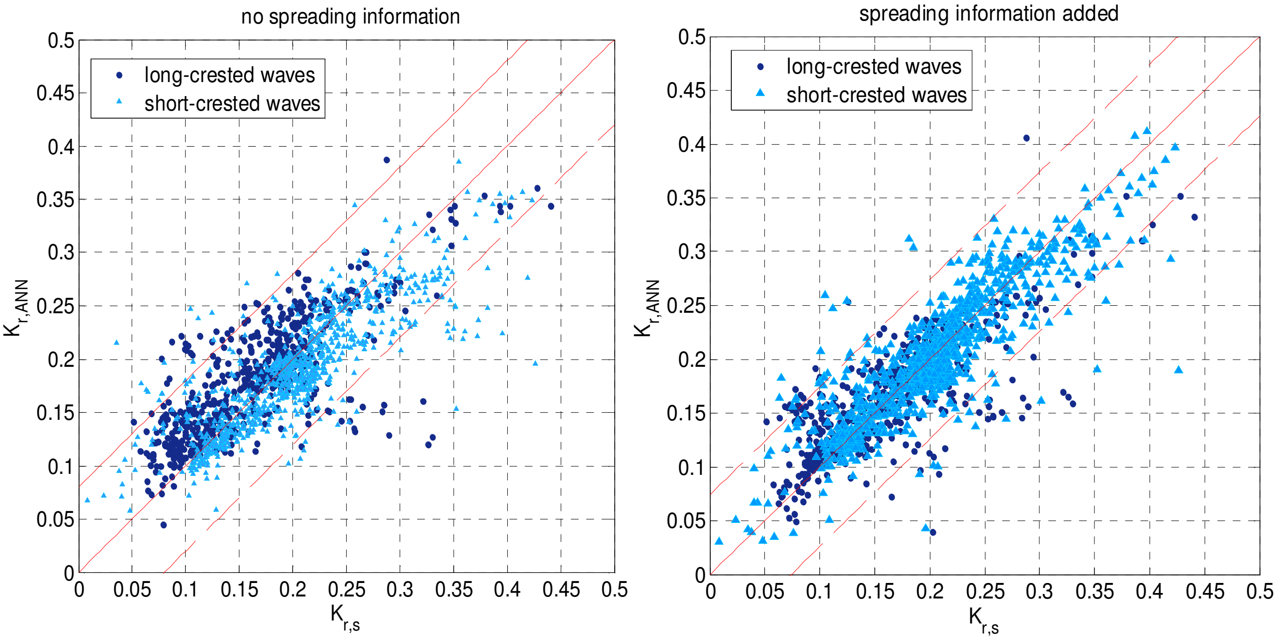


Figure 4.15 – Comparison among K_r predicted values ($K_{r,ANN}$, ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); to the left, prediction of the preliminary ANN trained without the spreading input parameter, to the right prediction including it.

The plot to the left of Figure 4.15 clearly denotes that the ANN could not deal with short-crested waves (i.e. *spreading* ≠ 0) as long as the wave directional spreading information was missing. With the exception of a few cases, all the $K_{r,s}$ values were systematically underestimated, data scatter under the continuous perfect fit line reveals. At the contrary, the inclusion of the *spreading* (Fig. 4.15 to the right) solved most of the shortcomings related to short-crested waves and improved the interpretation of long-crested waves. Moreover, it permitted to reduce the width of the confidence intervals.

4.5.1.4 Comparison with the existing formulae for the prediction of the wave reflection

Within the present paragraph, the comparison among the preliminary ANN performance and the traditional prediction formulae for the evaluation of K_r is provided. This kind of analysis represents an important element to discuss the ANN capability to overcome some of the traditional approaches shortcomings.

Due to limited ranges of validity associated to existing formulae, the comparison was managed against narrower datasets selected among the whole reflection database. Two cases were considered: perpendicular wave attacks on straight slopes and oblique wave attacks.

The preliminary ANN performance was compared with the results of Zanuttigh and van der Meer (2008) (hereafter, ZVDM formula, see Eq. 4.9) in case of straight slopes under perpendicular wave attacks.

$$K_r = \tanh\left(a \cdot \xi_0^b\right), \quad \text{where} \begin{cases} a = 0.167 \cdot \left[1 - \exp\left(-3.2 \cdot \gamma_f\right)\right] \\ b = 1.49 \cdot \left(\gamma_f - 0.38\right)^2 + 0.86 \end{cases} \quad \text{and} \quad \xi_0 = \frac{\tan\alpha}{\sqrt{\frac{2\pi H_{m,0,t}}{g T_{m-1,0}^2}}}$$

Eq. 4.9

By applying the ZVDM formula ranges of validity (i.e. design condition, $R_c/H_{m,0,t}$ ≥0.5, $H_{m,0,t}/D$ ≥1.0, $s_0 = H_{m,0,t}/L_{m-1,0,t}$ ≥0.01) to the whole database, a total of 724 data was selected (corresponding to part of the datasets A, B, C and D, see Tab. 4.3).

Figure 4.16 shows the comparison among the K_r values predicted by ANN and ZVDM as functions of $K_{r,s}$. Figure 4.17 displays the dispersion of the relative quantities $K_{r,ZVDM}/K_{r,s}$ and $K_{r,ANN}/K_{r,s}$ with increasing the relative crest freeboard $R_c/H_{m,0,t}$. In each diagram, the data are distinguished according to the type of armour (rock/unit) and core (permeable/impermeable).

Figures 4.16 and 4.17 show that the values obtained from the preliminary ANN, $K_{r,ANN}$ were characterized by a much lower dispersion around the ideal condition than the values $K_{r,ANN}$ (Fig. 4.17 to the left).

Furthermore, the ANN predictions seem not influenced by:

- the armour/core type, while ZVDM formula tends to systematically over-estimate K_r for rocks and permeable slopes and under-estimate K_r for rocks over impermeable core (Fig. 4.16);
- the relative crest freeboard, while ZVDM formula tends to systematically over-estimate K_r for $R_c/H_{m,0,t} < 2$ and to underestimate for $R_c/H_{m,0,t} > 3$ (Fig. 4.17).

Table 4.13 (left part, which is relative to straight slopes) reports the quantitative results of these simulations in terms of *rmse*, *WI* and R^2 values. The ANN allowed achieving more accurate predictions, even when applying it to the same range of data the ZVDM formula was based on.

Table 4.13 – *rmse*, *WI* and R^2 values obtained simulating ANN vs ZVDM formula (to the left) and ANN vs ZLA formula (to the right). Respectively, structures with straight slopes (left) and oblique wave attacks (right) only considered.

Preliminary ANN for the wave reflection				
	Straight slopes		Oblique wave attacks	
	ZVDM	ANN	ZLA	ANN
rmse	0.041	0.027	0.043	0.020
WI	0.925	0.973	0.816	0.971
R^2	0.75	0.89	0.49	0.89

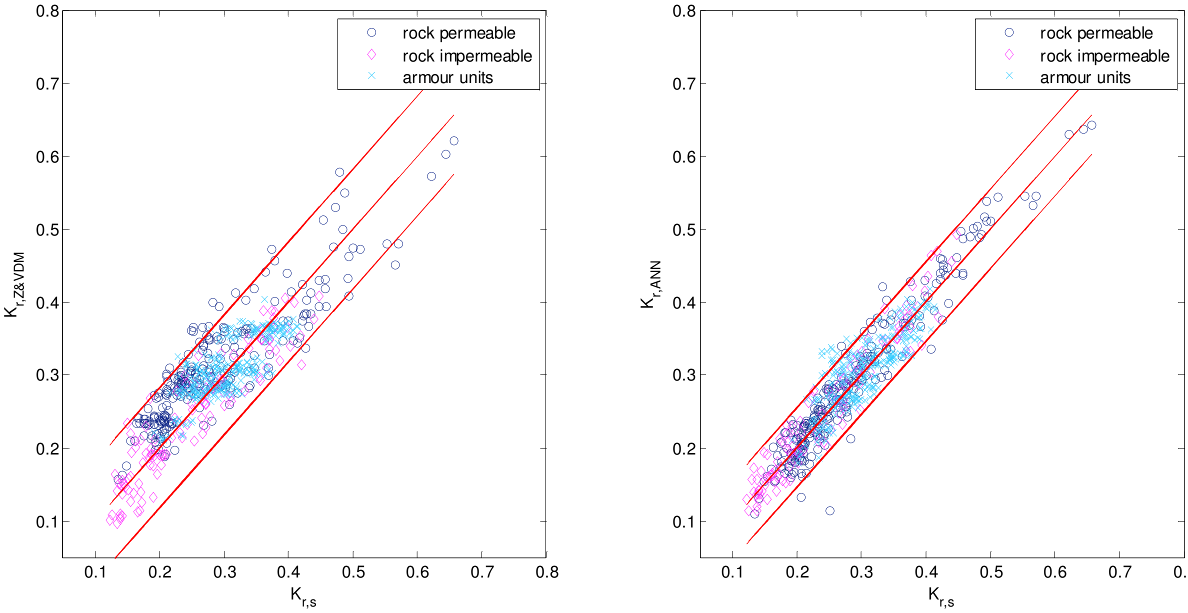


Figure 4.16 – Comparison among K_r predicted values ($K_{r,ANN}$, ordinate) and corresponding K_r experimental values (abscissa) for ZVDM formula predictions (to the left) and ANN predictions (on the right). Structures with straight slopes only considered. Preliminary ANN.

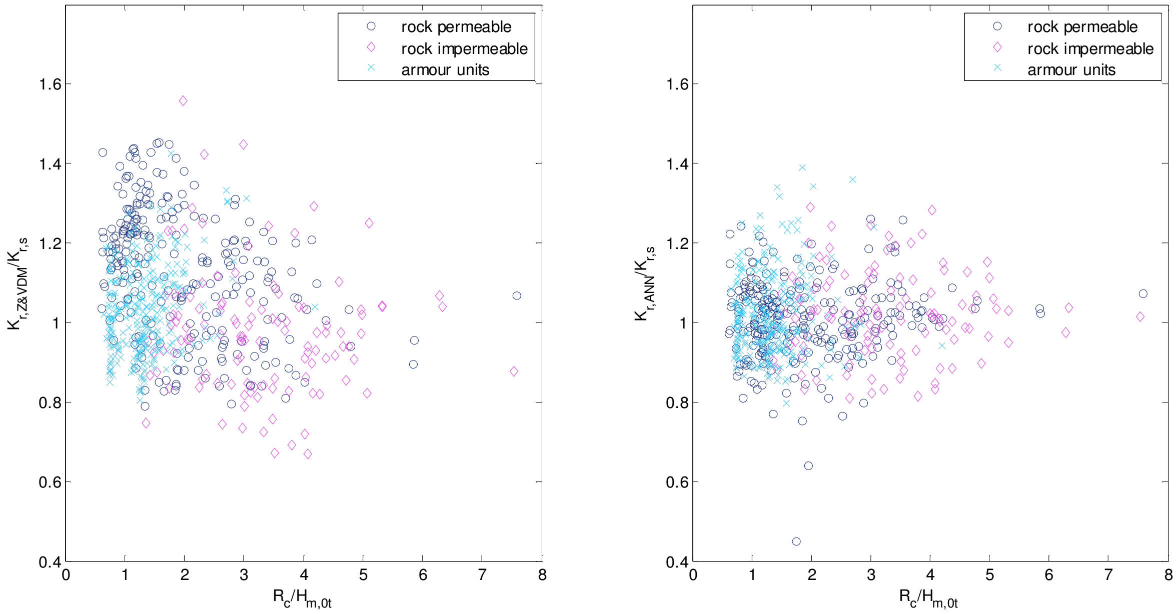


Figure 4.17 – Comparison among K_r predicted values in proportion to the corresponding K_r experimental ones (ordinate), as functions of the relative crest freeboard $R_c/H_{m,0,t}$ (abscissa) for ZVDM formula predictions (on the left) and ANN predictions (on the right). Structures with straight slopes only considered. Preliminary ANN.

In case of oblique wave attacks, the ANN performance was compared with the following formula of Zanuttigh and Lykke Andersen (2010), ZLA hereafter:

$$K_{r,ZLA}(\beta,s)=K_{r,ZVDM}(\beta=0,s)\cdot\gamma_{\beta},$$
$$where:\ \gamma_{\beta}=\begin{cases}(1\text{-}0.0077)\cdot\beta, & for\ long\text{-}crested\ waves\\(1\text{-}0.0058)\cdot\beta, & for\ short\text{-}crested\ waves\end{cases}'$$

Eq. 4.10

The expression of the wave obliquity factor γ_{β} in Eq. 4.10 is provided by Lykke Andersen and Burcharth (2009). The wave reflection coefficient $K_{r,ZVDM}$ is still given by Zanuttigh and van der Meer (2008) formula as above.

The comparison was performed over 736 tests throughout the dataset G (see Tab. 4.3) fulfilling the design conditions associated to ZVDM and ZLA formulae.

The comparisons among *rmse*, *WI* and R^2 values are reported in Table 4.13, part to the right. Again, as in the previous application of straight slopes, the preliminary ANN provided more accurate predictions even considering the same range of data the ZLA formula was based on.

The diagrams of Figures 4.18 and 4.19 aim to provide a qualitative analysis. The predicted values of K_r by the ANN and the ZLA formula are compared against experimental values in Figure 4.18; the dispersion of relative values $K_{r,ZLA}/K_{r,s}$ and $K_{r,ANN}/K_{r,s}$ are shown as functions of β in Figure 4.19. In these diagrams, short-crested and long-crested waves are also distinguished.

The ZLA formula tended to perform greater errors for greater values of β (Fig. 4.18), while the error computed by the preliminary ANN seemed not to be dependent on β. Both the ZLA formula and the ANN do not show any particular sensitivity to wave directional spreading (Fig. 4.19).

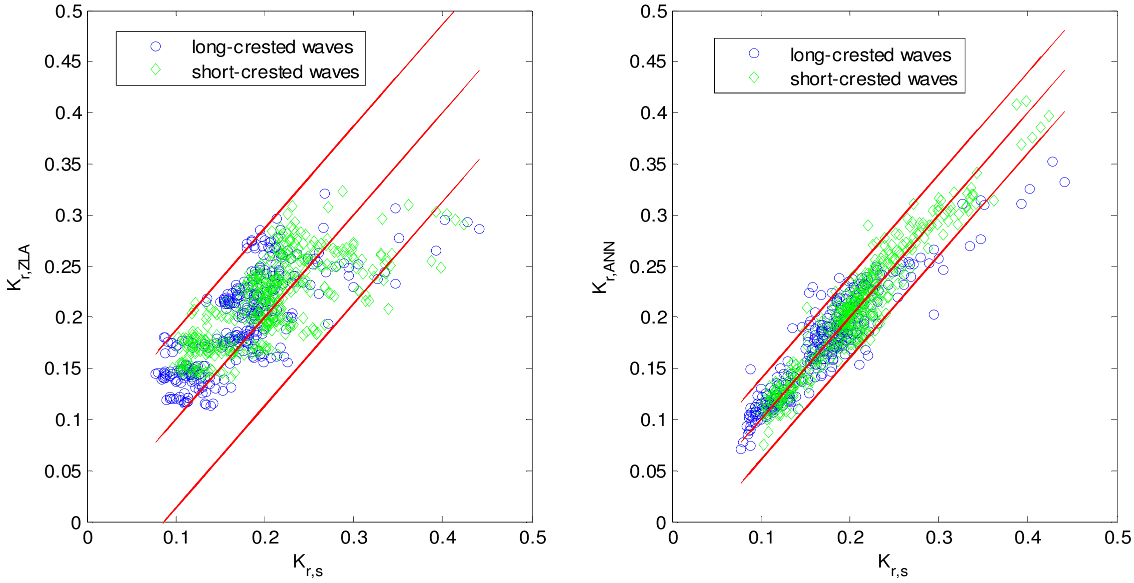


Figure 4.18 – Comparison among K_r predicted values ($K_{r,ANN}$, ordinate) and corresponding K_r experimental values (abscissa) for ZLA formula predictions (on the left) and ANN predictions (on the right). Oblique wave attacks only considered. Preliminary ANN.

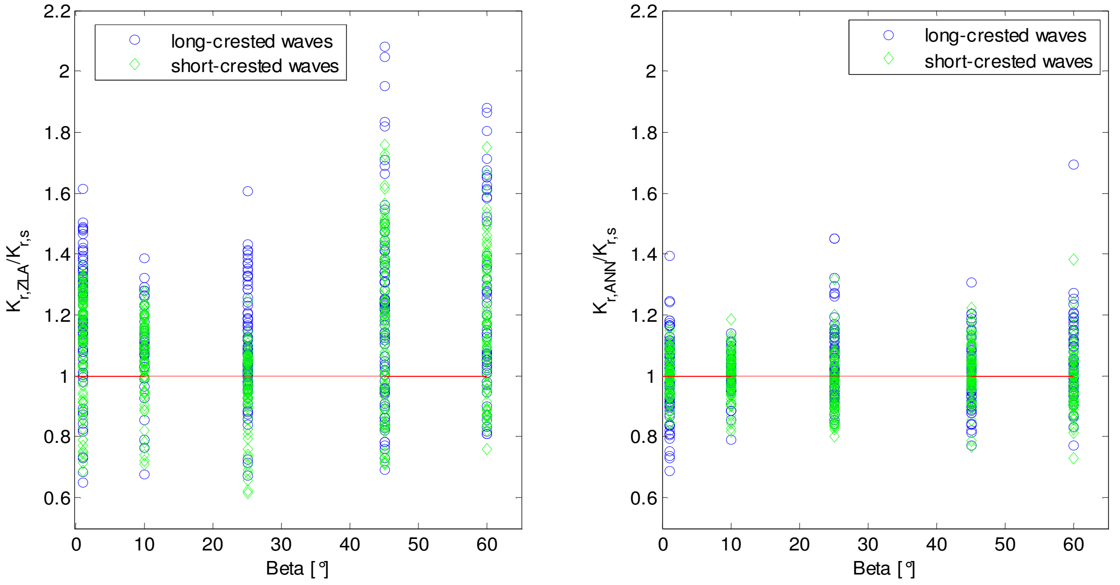


Figure 4.19 - Comparison among K_r predicted values in proportion to the corresponding K_r experimental ones (ordinate), as functions of the angle β (abscissa), for ZLA formula predictions (on the left) and ANN predictions (on the right). Oblique wave attacks only considered. Preliminary ANN.

4.5.2 Preliminary ANN for the wave transmission

This section presents and discusses the main results obtained by the preliminary ANN applied to the prediction of K_t.

The methodology of analysis of the results follows precisely the work already done for the application of the ANN to the wave reflection and here reported in Paragraph 4.5.1. Just a difference occurs between the two applications of the ANN: an additional routine (developed in *Matlab* language) was included for the representation of the wave transmission to prevent the prediction of negative values of K_t.

The performance of the preliminary ANN applied to the prediction of K_t is qualitatively shown by the comparison of $K_{t,ANN}$ values with $K_{t,s}$ (Fig. 4.20, to the left) and by the dispersion of the absolute error $e = K_{t,s} - K_{t,ANN}$ as a function of $K_{t,s}$ (Fig. 4.20, to the right). A quantitative estimate of the ANN accuracy is provided by the average values (resumed in Tab. 4.13) of the usual three error indexes (*rmse*, *WI* and R^2), and the corresponding standard deviations. To allow an easier comparison, Table 4.14 reports also the error indices associated to:

- the ANN trained on the narrower database of 2'285 data (the same employed by Panizzo and Briganti, 2007, hereafter, PB and $\mathrm{ANN_{PB}}$);
- the existing transmission $\mathrm{ANN_{PB}}$, where the $rmse_{\mathrm{PB}}$ value has been derived from the Dataset 07 reported in Tab. 1 of PB;
- the preliminary reflection ANN (see Paragraph 4.5).

In order to assess the uncertainty associated to the performance of the model, the results were derived as average values obtained from several different train and run of the ANN. The stochastic independence of the results was guaranteed by the re-initialization of the ANN performed before any training process and by the employment of a bootstrap technique to resample the training database each time. Please, note that the adoption of the bootstrapping technique in substitution of the "early-stopping" was not yet applied at this stage of the work.

A specific sensitivity analysis revealed that a number of 20 training-testing-simulation processes was sufficiently large to describe the actual uncertainty of the ANN error distribution fully. This value of the optimal number of simulations is sensibly lower than for reflection (50), probably due to the greater extension of the wave reflection database (5'781 data against 3'379), and therefore to the wider ranges of values of the 13 input elements.

Table 4.14 - *rmse, WI* and R^2 average values and corresponding standard deviations derived from the 20 simulations; transmission ANN (trained on the complete database of 3379 and on the 2285 data employed by PB) is compared to reflection ANN and $\mathrm{ANN_{PB}}$ (data refer to Set07 in Tab. 1 of PB work).

	ANN for K_t (#3379)	ANN for K_t (#2285)	$\mathrm{ANN_{PB}}$ for K_t (#2285)	ANN for K_r (#5781)
rmse	0.037 ± 0.003	0.033 ± 0.002	0.065	0.038 ± 0.003
WI	0.993 ± 0.001	0.9941 ± 0.0006	-	0.985 ± 0.003
R^2	0.973 ± 0.004	0.978 ± 0.002	0.983	0.943 ± 0.006

Within the present application for the estimation of K_t, the preliminary ANN showed the tendency to produce some negative values of the predicted coefficient $K_{t,ANN}$. On average 25 values $K_{t,ANN} < 0$ occurred at each simulation, especially in correspondence of very low experimental values $K_{t,s}$). To eliminate any negative value of $K_{t,s}$, a routine which reads the predictions performed by the ANN after each simulation and substitutes each negative predicted value with with a "*NaN*" ("Not a Number", in *Matlab* language) was developed. The addition of this routine in the code represented the only modification applied to the preliminary ANN.

From both the graphs of Figure 4.20, it can be appreciated the very good agreement of computations and measurements and above all the great degree of symmetry in the error values distribution. Both the highest and the lowest values of K_t are pretty well represented by the ANN, which therefore did not appear to be affected by systematic errors. Just a few scattered values out of 95% confidence level bands (dashed lines) are detectable.

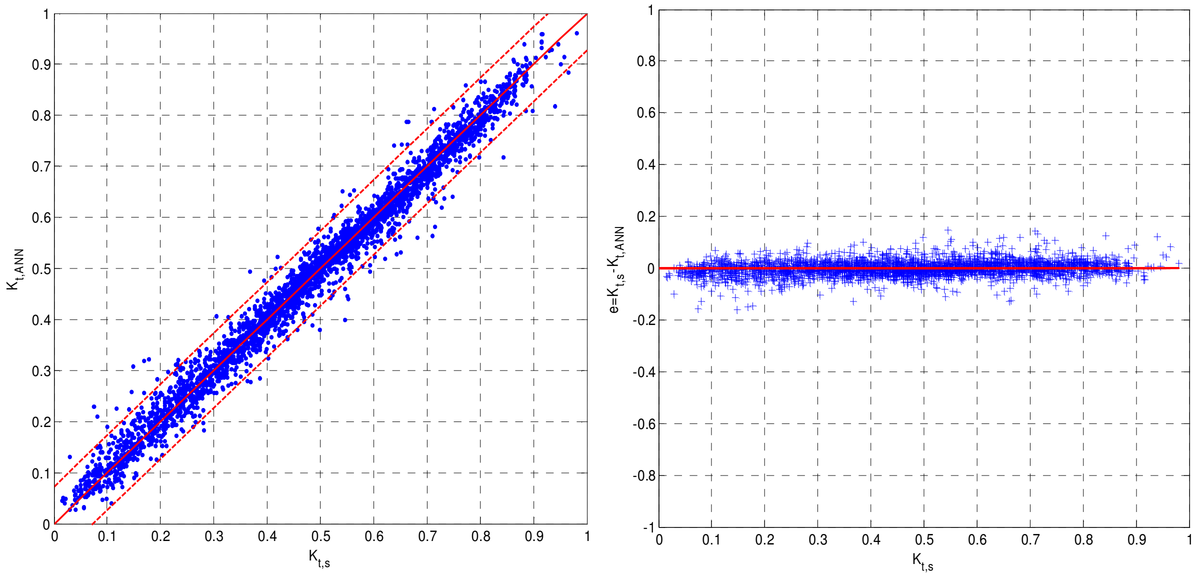


Figure 4.20 – Left: Comparison among K_t predicted values ($K_{t,ANN}$ ordinate) and corresponding K_t experimental values ($K_{t,s}$, abscissa); the continuous bisector represents the ideal condition ($K_{t,ANN}$ = $K_{t,s}$), while the dashed lines refer to the 95% confidence levels. Right: difference $e = K_{t,s} - K_{t,ANN}$ (ordinate) as a function of $K_{t,s}$ (abscissa).

From a quantitative viewpoint, the *rmse* value (0.037, referring to the complete database, see Tab. 4.14) is slightly lower than the one which characterizes the prediction of K_r (0.038). It represented a satisfactory result, either in comparison to ANN$_{PB}$ (see more details in following Section 4.5.2.1), either considering that the preliminary ANN architecture was optimized for wave reflection and not for wave transmission.

Similarly, the very high value of *WI* (larger than 0.990) confirms the great symmetry in the errors distribution qualitatively appreciated in the diagrams of Figure 4.20. Also R^2 value is greater than the value associated to the prediction of K_r (0.985).

The values of the standard deviation associated to the average indexes were computed to assess the uncertainty of the errors and quantify the ANN stability: consistently with the wave reflection case, each index is characterized by a standard deviation value of about 10^{-3}.

Similarly to the application case of the wave reflection (see Paragraph 4.5.1), a specific analysis of the errors was carried out, in order to detect and define the “largest” errors computed by the ANN, which are not supposed to fall within the random uncertainty. The “threshold” value $e \geq |0.15|$ (i.e. $e_{large} \overset{\text{def}}{=} e \geq |0.15|$) previously identified for K_r was employed again.

Similarly to Fig. 4.10, all the “large” errors computed during the 20 simulations are shown in Figure 4.21 as a function respectively of the progressive test indices (left diagram) and of $K_{t,s}$ (right diagram). The concentration of “large” errors associated to a particular test (plot to the left) or to a particular value of $K_{t,s}$ (plot to the left) are therefore immediately visible. If the same test is systematically affected by “large” error, it may imply either that the test itself is less “reliable” (for example, due to error measurements, especially for the lowest and the greatest values of K_t), or that the ANN is not able to correctly represent that test.

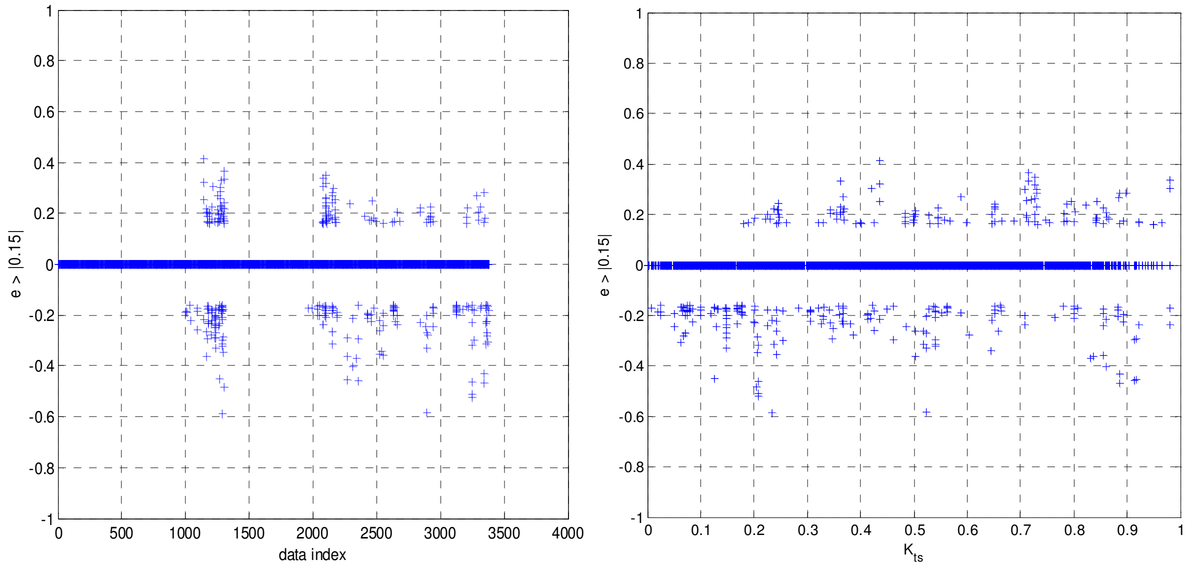


Figure 4.21 – Left: $|e| = |K_{t,s}\text{-}K_{t,ANN}| \geq 0.15$ (ordinate) as a function of database test indexes (abscissa); right: $|e| = |K_{r,s}\text{-}K_{t,ANN}| \geq 0.15$ (ordinate) as a function of $K_{t,s}$ (abscissa).

By observing the plot to the left of Figure 4.21, it is clear that some datasets were never affected by "large" errors: in particular, the dataset identified by indices values < 1000, referring to aquareefs, and the dataset associated to 1300÷2000, referring to rock LCSs collected by Seabrook and Hall (1998). By observing the plot to the right of Figure 4.22 it is clear as well that the ANN captured with similar accuracy both high or low values of K_t, since the "large" errors distribution is randomly spread around the whole range of K_t values.

The results of this qualitative analysis might lead to the conclusion that "large" errors occur especially for specific datasets, which correspond to oblique and 3D wave attacks (indices 1'000÷1'300), Melito and Melby (2002), tests (indices 2000÷2100) and Daemrich, et al. (2001) tests (indices > 3'200).

To provide a quantitative estimate of the frequency of occurrence of "large" errors and an assessment of the "worst" condition of prediction, the same indexes of Eq. 4.7 and 4.8 have been applied also to this application:

$$\bar{e}_{\%} = \frac{\bar{e}}{K_{t,s}} \cdot 100, \;\; where\; \bar{e} = \frac{1}{20} \cdot \sum_{i=1}^{20} \left(\frac{1}{3379} \cdot \sum_{j=1}^{3379} |e_{j,i}| \right)$$

Eq. 4.11

$$\bar{e}_{large,\%} = \frac{\bar{e}_{large}}{\overline{K}_{t,s}} \cdot 100, \;\; where\; \bar{e}_{large} = \frac{1}{308} \cdot \sum_{k=1}^{308} |e_{large}|$$

Eq. 4.12

In Eq. 4.12, '308' is the total number of "large" errors computed over the 20 simulations of the ANN, i.e. over more than 67'000 data.

The indexes $\bar{e}$ and $\bar{e}_{\%}$ (Eq. 4.11) respectively represent the absolute and the relative errors computed on average by the ANN, while the indexes $\bar{e}_{large}$ and $\bar{e}_{large,\%}$ (Eq. 4.12) represent the same average quantities computed considering the 308 "large" errors only. The values of all the indexes are resumed in Table 4.15: the "worst" condition is represented by $\bar{e}_{large,\%}$ = 48% i.e. the ANN could produce errors equal or larger than 48% with a frequency of 0.4% (just 4 times over 1'000 data).

Table 4.15 - Values of the average absolute and relative errors provided by the preliminary ANN over the 20 simulations.

Preliminary ANN for the wave transmission				
$\bar{e}$	$\bar{e}_{\%}$	$\bar{e}_{large}$	$\bar{e}_{large,\%}$	$\bar{K}_{t,s}$
0.020	4.3%	0.223	48%	0.462

4.5.2.1 Comparison with the existing transmission ANN

The aim of this section is to provide a comparison between the performance of the preliminary ANN and ANN_{PB}, both referring to the evaluation of K_t. ANN_{PB} was trained against 2'285 test which were all included in the training dataset of the preliminary ANN.

The comparison is qualitatively given by the analysis of the respective diagrams $K_{t,ANN}$ vs $K_{t,s}$ (plot of Fig. 7a in PB paper, Fig. 4.20 to the left in the present work) and quantitatively by the respective values of *rmse* and R^2 indexes (Tab. 4.14). Both the ANNs show a narrow distribution around the ideal condition (represented by the bisector of the diagrams), however the scatter of the predicted values $K_{t,ANN}$ by ANN_{PB} is greater than the one obtained by the preliminary ANN, especially for $K_{t,s} > 0.85$.

Regarding the error indexes, ANN_{PB} is characterized by $rmse_{PB}$ = 0.063 (see Tab. 4.14) and R^2_{PB} = 0.983, denoting generally a similar performance (in effect, $rmse_{PB}$ suggests a worse performance, while at the same time R^2_{PB} a better one).

The results of the preliminary ANN were therefore satisfactory, also considering that ANN_{PB} architecture was specifically calibrated against wave transmission tests and trained over a narrower database. For example, the updated transmission database on which the preliminary ANN was trained contained 299 tests related to smooth structures, which were excluded in ANN_{PB}.

To investigate more in depth the role played by the additional data, the preliminary ANN was trained and simulated against the same narrowest dataset employed by PB (see Tab. 4.14). All the indexes showed in this case an improved performance, also remarked by the decrease of the values of the standard deviations, which drop even below 10^{-3}.

Following the work of PB, a further analysis was carried out to discuss the distributions of the errors (the quantity *e*) and of the predicted values $K_{t,ANN}$ as functions of some specific input elements, such as the relative crest freeboard $R_c/H_{m,0,t,}$ and the non-dimensional structure crest width $G_c/L_{m-1,0,t}$. These error distributions are shown in Figure 4.22, where the plot to the left ($R_c/H_{m,0,t}$ on the abscissa) is compared with Figure 8a in PB, while the plot to the right ($G_c/L_{m-1,0,t}$ on the abscissa) with Figure 9a in PB. Both the present diagrams show a noteworthy reduction of the scatter.

In the left plot of Fig. 4.22, *e* increases up to and over |0.1| just within $R_c/H_{m,0,t}$,= [-1;1] while, in the corresponding PB diagram, errors lower than -0.1 are detectable within the whole range of $R_c/H_{m,0,t}$.

In the right panel, the distribution of *e* is symmetric and narrower around the ideal condition *e* = 0 for each value of $G_c/L_{m-1,0,t}$. More in details, Fig. 9a in PB shows some amount of points gathered around a specific value of $G_c/L_{m-1,0,t}$, while in the present work (Fig. 4.22 to the right) the points are more smoothly distributed on the abscissa, i.e. the error distribution is more uniform over the range of $G_c/L_{m-1,0,t}$.

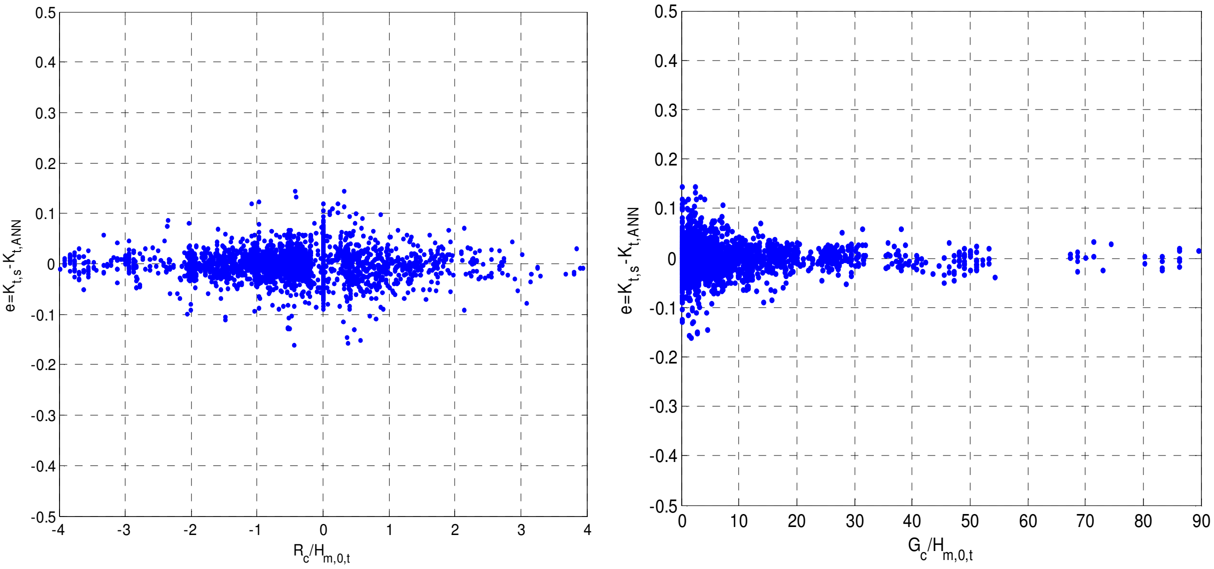


Figure 4.22 - Left: $e = K_{t,s} - K_{t,ANN}$ (ordinate) as a function of the relative crest freeboard $R_c/H_{m,0,t}$ (abscissa); right: $e = K_{t,s} - K_{t,ANN}$ (ordinate) as a function of the non-dimensional structure crest width $G_c/L_{m-1,0,t}$.

Figure 4.23 presents K_t as function of $R_c/H_{m,0,t}$ ($K_{t,s}$ values on the ordinate to the left panel, $K_{t,ANN}$ to the right), and corresponds to the panels (a) and (b) of Figure 10 in PB paper. The experimental distribution is pretty well reproduced by the predicted values of the preliminary ANN, even taking into account the lowest and the highest values of $R_c/H_{m,0,t}$.

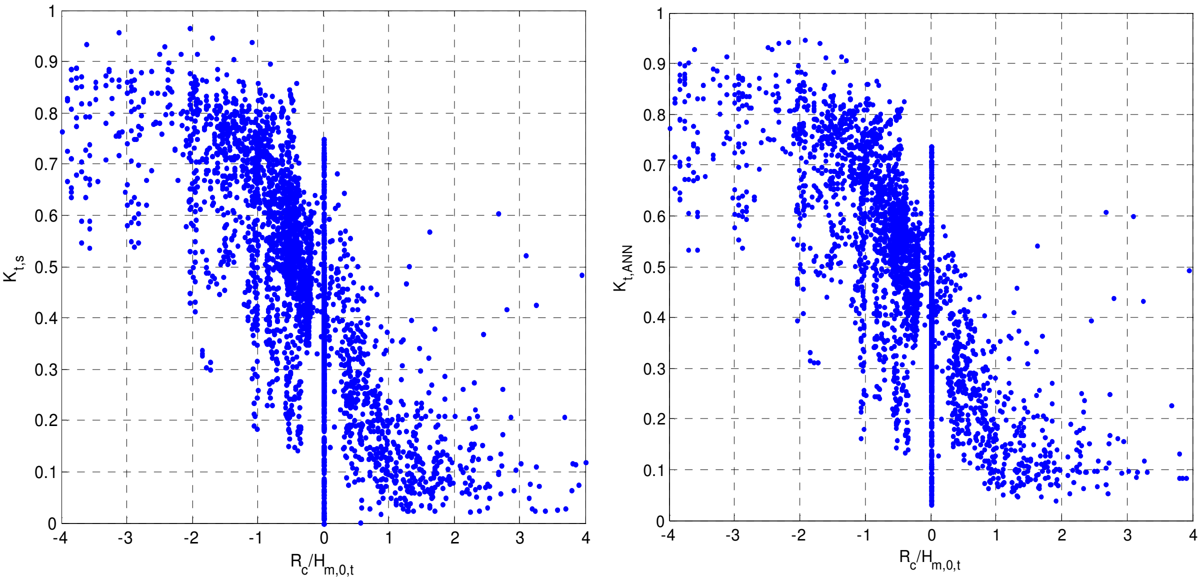


Figure 4.23 – $K_{t,s}$ (ordinate, left panel) compared to $K_{t,ANN}$ (ordinate, right panel) as functions of $R_c/H_{m,0,t}$ (abscissa).

The dependence of $K_{t,s}$ and of $K_{t,ANN}$ on $G_c/L_{m-1,0,t}$ is reported respectively to the left and to the right of Figures 4.24 and 4.25. Similarly to PB, the data have been divided in classes of $R_c/H_{m,0,t}$ and only the structures with $R_c/H_{m,0,t} = 0$, i.e. the ones showing the greater scatter, have been used in this analysis. The data have been distinguished according to the values of the armour stone diameter ($H_{m,0,t}/D$) in Figure 4.24 and according to the values of the breaking parameter ξ_0 in Figure 4.25.

By comparing the plots to the right of Figures 4.24 and 4.25 respectively with the panels (a) and (b) of Figure 12 of PB, two main issues are immediately detectable:

- the preliminary ANN presented in this work shows an improved predicting capacity, especially for high values of K_t. While ANN$_{PB}$ never predicts K_t values approximately greater than 0.55, the preliminary ANN can correctly reproduce also values greater than 0.7;
- the preliminary ANN is instead not able to reproduce K_t values lower than 0.02÷0.03, probably due to the routine developed to prevent negative predictions (see Paragraph 3.2). More precisely, the minimum vale $K_{t,ANN} = 0.027 \approx 0.03$ can be defined as the lower minimum of validity of the ANN. This problem – which did not affect ANN$_{PB}$ – is particularly evident for $G_c/L_{m-1,0,t} > 0.40$ (compare left and right panels of both Fig.s. 4.24 and 4.25).

Therefore, if by one hand ANN$_{PB}$ seems to be upper-limited (≈ 0.55), by the other hand ANN is lower-limited by ≈ 0.03.

Likewise ANN$_{PB}$, the preliminary ANN overcomes the discontinuity of Van der Meer et al. (2005) formulae (see Eq. 4.13) for $G_c/L_{m-1,0,t} = 0$ and is able to represent the dependence on D and ξ_0.

$$K_t=-0.4\frac{R_c}{H_{m,0,t}}+0.64\left(\frac{G_c}{H_{m,0,t}}\right)^{-0.31}\left(1-e^{-0.5\xi_{o,p}}\right),\quad if\ \frac{G_c}{H_{m,0,t}}<10$$
$$K_t=-0.35\frac{R_c}{H_{m,0,t}}+0.51\left(\frac{G_c}{H_{m,0,t}}\right)^{-0.65}\left(1-e^{-0.41\xi_{o,p}}\right),\quad if\ \frac{G_c}{H_{m,0,t}}>10$$

Eq. 4.13

Furthermore, the experimental values which do not follow Van der Meer distribution (see left panels of Figs. 4.24 and 4.25) are pretty well reproduced by the here presented ANN (corresponding right panels). On the contrary, the values $K_{t,ANN,PB}$ are generally aligned with Van der Meer predictions (panels (a) and (b) in Fig. 12, PB paper), showing not significant improvement with respect to the formulae.

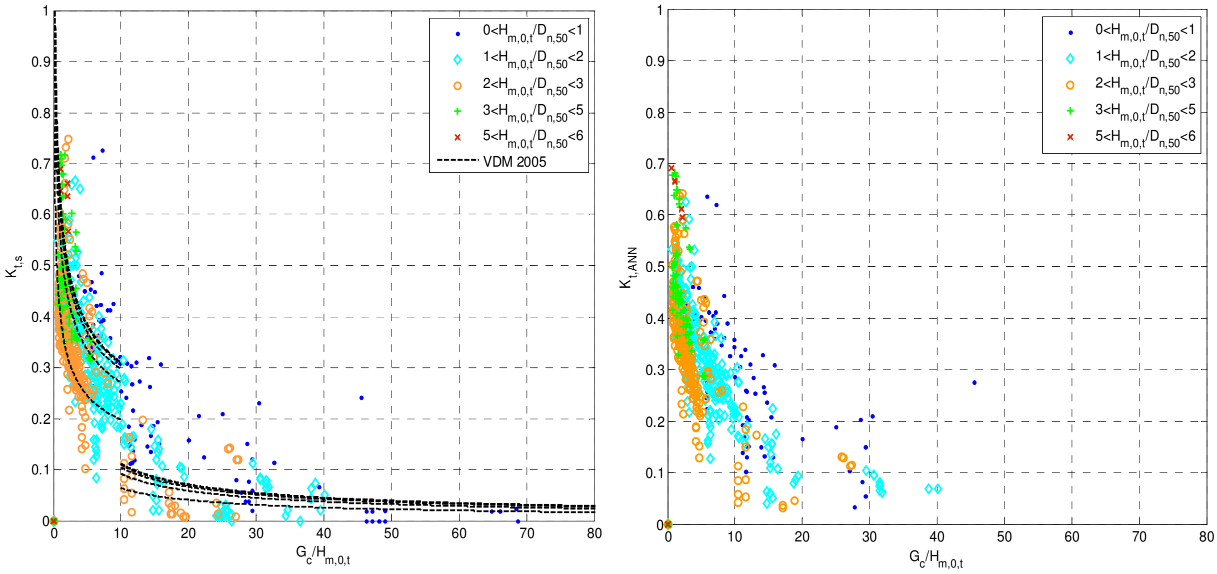


Figure 4.24 - $K_{t,s}$ (ordinate, left panel) compared to $K_{t,ANN}$ (ordinate, right panel) as functions of $G_c/L_{m-1,0,t}$ (abscissa) at different classes of $H_{m,0,t}/D$. Only data characterized by $R_c/H_{m,0,t} = 0$ are included.

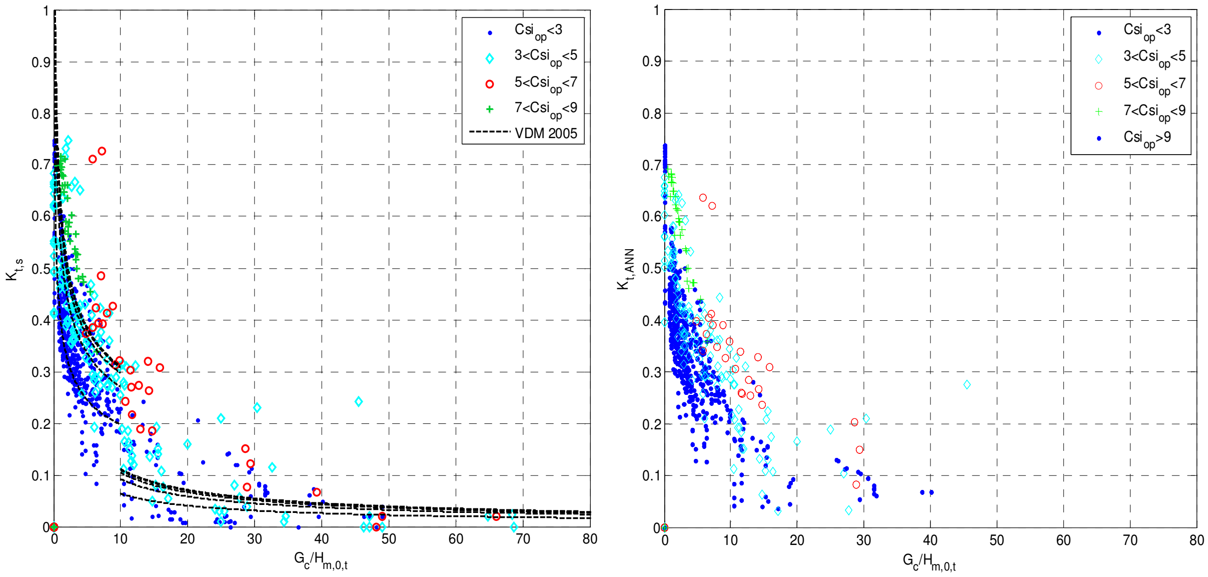


Figure 4.25 - $K_{t,s}$ (ordinate, left panel) compared to $K_{t,ANN}$ (ordinate, right panel) as functions of $G_c/L_{m-1,0,t}$ (abscissa) at different classes of ξ_0. Only data characterized by $R_c/H_{m,0,t} = 0$ are included.

4.5.2.2 Discussion about the preliminary ANN applied to wave transmission

The preliminary ANN optimized for the wave reflection and here applied to the wave transmission generally demonstrated an improved performance with the respect to ANN$_{PB}$, considering both the indexes and the distribution of the errors. However, it is worthy to discuss the following key issues that may affect the preliminary ANN applicability (and therefore the applicability of the new ANN itself):

- the number of input parameters required (13 instead of the 6 needed by ANN$_{PB}$, in case of the preliminary ANN and even 16 for the new ANN), which means that more detailed experimental/prototype information is needed to run the ANN;
- the number of the hidden neurons (40 instead of 6, for both preliminary and new ANNs), which represents an increased complexity of the ANN architecture, and requires extended databases to correctly train and calibrate the model;
- the limitation in representing low values: $K_{t,ANN} < 0.03$; however, this limit has been overcome by the new optimized ANN).

It has to be noted that the high dimension of the hidden layer may affect the ANNs applicability only during its training, i.e. only in case the model has to be re-initialized. As one ANN is already trained, it is suitable to be directly applied at least to similar LCSs, and therefore this issue does not affect anymore the ANN.

The limit of 0.03 in most cases is lower than the measurements uncertainty. The upper limit of 0.55 is instead a more restrictive condition in ANN$_{PB}$, since the number of tests presenting $K_{t,s} > 0.55$ is not negligigle (see Tab. 4.16).

Table 4.16 - Number of tests presenting $K_{t,s} < 0.03$ and $K_{t,s} > 0.55$ within the complete database (3'379 data) and PB database (2'285 data) and relative percentages with the respect of the total number of data.

	ANN # 3379	**ANN$_{PB}$ # 2285**
$K_{t,s} < 0.03$	61; 2%	44; 2%
$K_{t,s} > 0.55$	1266; 37%	890; 39%

5. THE RESULTS OF THE NEW ANN: THE PREDICTION OF THE WAVE OVERTOPPING DISCHARGE, THE WAVE TRANSMISSION AND THE WAVE REFLECTION COEFFICIENTS

5.1 Introduction and outline

The outcomes of the “new” advanced ANN are presented in this Chapter.

The performance of the ANN – in reference to all the output parameters, q, K_r and K_t – is separately discussed for each process in a first Paragraph 5.2. The quantitative analysis of performance is provided by means of three error indexes computed among experimental and predicted values: the root mean square error (*rmse*), the Willmott index (*WI*) and the coefficient of determination (R^2). The results are qualitatively investigated through plots and diagrams that report the comparison among experimental and predicted values and the distribution of the error computed by the ANN.

A separate Paragraph (5.3) is dedicated to illustrate the ANN performance against existing ANNs (5.3.1) and the main formulae available in literature (5.3.2). For these comparisons, the ANN is alternatively applied to specific datasets of the overall database, according to the field or ranges of validity of the formulae.

Besides, an attempt to provide “contemporary” predictions of more than one output parameter is shown and discussed in Paragraph 5.4. Aim of this section is essentially to show and stress one of the limits of the ANN, or, better, of the so far available database and to point out the main steps for further research.

The final Paragraph 5.5 is focused on the implementation of a “logic” classifier for the optimization of the prediction of the wave overtopping discharge. This Paragraph will show how the adoption of a “logic” classifier (similar to the work by Verhaeghe et al. (2008) for the developed for the CLASH ANN) does not lead to any particular improvement of the ANN. This issue link the present chapter to the next one (Chapter 6), where a new methodology for the prediction of both “large” and “small” values of overtopping discharge will be proposed.

5.2 The performance of the ANN

The final layout of the ANN has been defined as a result of the sensitivity analysis to the input parameters, the number of hidden neurons and the other characteristic features of the ANN (see Chapter 4). This section aims to present the main results obtained by the application of the “optimized” new ANN to the prediction of q, K_r and K_t. The performance of the ANN has been investigated by means of the same numerical error indexes and the so-called “large errors” already defined in Chapter 4 (see, into specific, Paragraph 4.3 and Eq.s 4.3, 4.4, 4.5 and 4.6).

The quantitative results of the predictions of q, K_r and K_t are summarized in Table 5.1 in terms of the average values derived after 50 bootstrap resamples of the ANN. The choice to develop 50 resamples of the database is related to the assessment of the performance and has been already discussed in Paragraph 4.4.1. Within Tab. 5.1, the effective number of data involved in the training is reported for each application case. These numbers are all lower than the overall number of tests included in each of the corresponding databases (see Paragraph 4.2), since the tests reporting *RF* or *CF* = 4 have been discarded (see Paragraph 4.4.4).

As it can be observed from Tab. 5.1, the *rmse* value associated to the prediction of q presents the same order of magnitude of the other outputs (i.e. 10^{-2}), while in the work by Van Gent et al (2007) it was ~ 10^{-1}. In fact, in the present research, the experimental values of q were subjected to a further transformation with respect to the logarithmic scale and the non-dimensional group $(g\cdot H_{m,0,t}{}^{3})^{0.5}$ presented in Eq. 4.6 and previously adopted by Van Gent et al (2007).

The target output for the present ANN is actually the quantity q^* computed as in eq. 5.1:

$$q^*=\frac{log_{10}(q_{AD})\text{ - }min\{log_{10}(q_{AD})\}}{|\ min\{log_{10}(q_{AD})\}\text{ - }max\{log_{10}(q_{AD})\}|},$$

Eq. 0.1

where the experimental values of q are still non-dimensionalized to the quantity q_{AD}, by means of the factor $(g\cdot H_{m,0,t}{}^{3})^{0.5}$, as shown in Eq. 5.2.

$$q_{AD}=\frac{q}{\sqrt{gH_{m,0,t}^{3}}}$$

Eq. 0.2

The non-dimensional quantity q_{AD} is then transformed into logarithmic scale and finally q^* is obtained by scaling q_{AD} with respect to its maximum and minimum and translating it to the range [0; 1] (see eq. 5.1). Actually, the ultimate aim of the transformation from q to q^* is to obtain target values of q between 0 and 1, similarly to the natural values of K_r and K_t. This objective is pursued to uniform the results and ease the comparison of the performance among the three application cases. Besides, the employment of q^* has revealed to allow a slight improvement of the performance, as already demonstrated and pointed out by Zanuttigh et al. (2014). However, it has to be clarified that the transformation applied to the target values before training the ANNs (Eq.s 5.1 and 5.2) is completely undone before supplying the results to the user.

Table 0.1 – Synthesis of the performance of the ANN in its final layout. Average results from 50 bootstrap resamples of the database.

Prediction of the three outputs: K_r, K_t and q				
Output (# nr. of tests)	*RMSE*	*WI*	R^2	*# large errors* *(%)*
$q \geq 10^{-6}$ m^3/(sm) (#7'716)	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
K_r (#5'824)	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
K_t (#2'888)	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%

The numerical values presented in Tab. 5.1 show that the predictions of K_r and K_t are generally more accurate (see the corresponding values of *rmse* and R^2) and the distribution of the error more symmetric (see *WI*) with respect to q. However, the standard deviations associated to the error indexes and the number of large errors are greater for K_r and K_t. These results lead to the conclusion that the ANN performs an "average" more precise prediction for K_r and K_t, but the prediction is less "stable". In other terms, the prediction of K_r and K_t is more sensible to the resampled database selected for the training at each run of the ANN. Therefore, the choice to discard the early stopping technique and adopt the methodology of the "commitment of networks" – suggested by Verhaeghe (2005) and actualized in the implementation of the bootstrap resampling technique (see Paragraph 4.4.3) – is more significant for these outputs than for q. This aspect is remarked by the comparison of performance obtained with and without early stopping provided in Tab. 4.8: the differences between the two approaches are sharper for K_r and K_t.

For this reason, it is particularly important that the outcome of the ANN be an "average" prediction: this implies that the ultimate tool to be delivered to the users is represented by 50 ANNs, each of them trained on a different database. Each ANN is supposed to process the input parameters defined by the user and predict the output separately; therefore, the 50 output estimations are averaged and a mean result is delivered, together with the confidence interval associated to the prediction.

The discussion about the ANN performance is more in-depth and singularly investigated for each process in the following Paragraphs, with the support of qualitative plots and diagrams.

5.2.1 Wave reflection

The plots of Figure 5.1 aim to illustrate the ANN performance in terms of comparison among computed values of K_r, i.e. $K_{r,ANN}$, and measured ones, $K_{r,s}$ (diagram to the left) and dispersion of the absolute error $e= K_{r,s} - K_{r,ANN}$ against $K_{r,s}$ (diagram to the right). The central line in the plot to the left represents the bisector, i.e. the perfect correspondence among predicted and experimental values, while the external lines represent the 95% confidence boundaries.

These plots are completely similar to the ones of Fig. 4.8, relative to a preliminary ANN. By comparing the diagrams to the left of Fig. 5.1 and Fig. 4.8, an increased number of tests (Fig. 5.1, to the left), mainly referring to the high values of $K_{r,s}$ can be appreciated. Indeed, the wave reflection database was extended after the implementation of the preliminary ANN, collecting an overall number of 7’413 data against the original 5’781 (compare Fig. 4.3 with the database described in Zanuttigh et al., 2013).

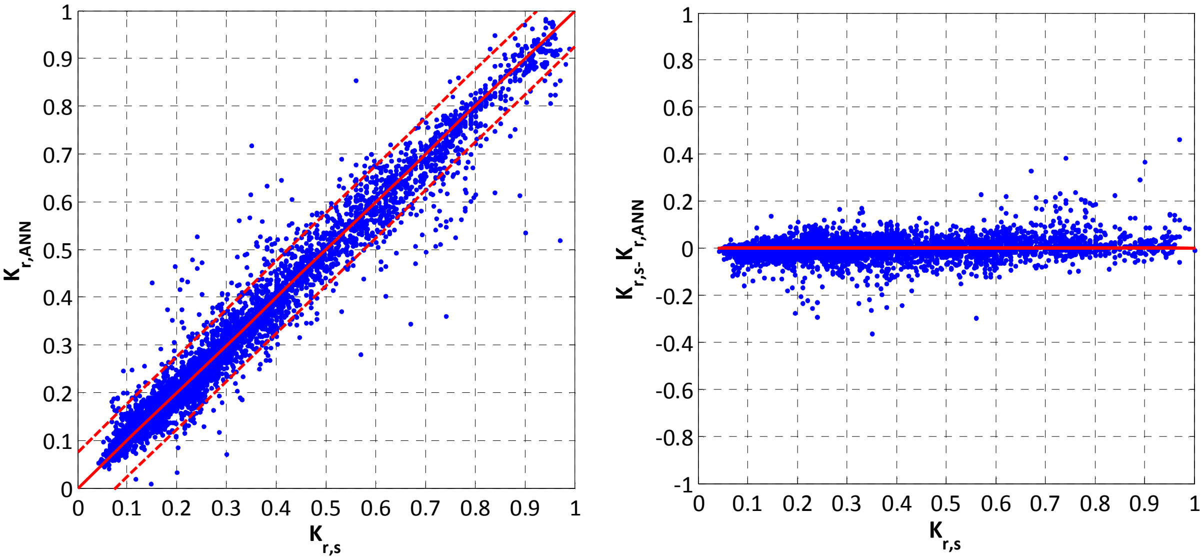


Figure 0.1 – Left: Comparison among K_r predicted values ($K_{r,ANN}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); the continuous bisector represents the ideal condition ($K_{r,ANN} = K_{r,s}$), while the dashed lines refer to the 95% confidence levels. Right: difference $e = K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of $K_{r,s}$ (abscissa). New advanced ANN.

The extension of the database, from one hand, has leaded to a reduction of the bias for high values of $K_{r,s}$, but, from the other hand, to an increase of the general scatter, as the presence of isolated values out of the 95% error bands (Fig. 5.1 left) enlightens.

In order to individuate the tests more “responsible” of the greater scatter, a sensitivity analysis of the error distribution to the main physical parameters involved in the phenomenon of the wave reflection was been carried out. The individuated key parameters are essentially the relative crest freeboard $R_c/H_{m,0,t}$ and the wave steepness $H_{m,0,t}/L_{m-1,0,t}$. These quantities proved to play a key role in the characterization of all the wave-structure interaction processes (for K_r, see Zanuttigh and van der Meer, 2008; for K_t, see van der Meer et al., 2005; Panizzo and Briganti, 2007; for q, see EuroTop, 2007; van der Meer et al., 2013) and for this reason were employed also for the error analysis of the wave transmission and the wave overtopping and.

Following this approach, the diagrams of Fig.s 5.2 and 5.3 were realized. In both the figures, the predicted data are distinguished according to the different belonging section of the database. The seven sections – already described and reported in the diagram of Fig. 4.3 – subdivide the database into classes of tests characterized by the same structure type or wave attack conditions.

In case of Fig. 5.2, the dispersion of the error $e = K_{r,s}$ - $K_{r,ANN}$ is reproduced as a function of the relative crest freeboard $R_c/H_{m,0,t}$, while in Fig. 5.3 the same dispersion is plotted as a function of the wave steepness $H_{m,0,t}/L_{m-1,0,t}$.

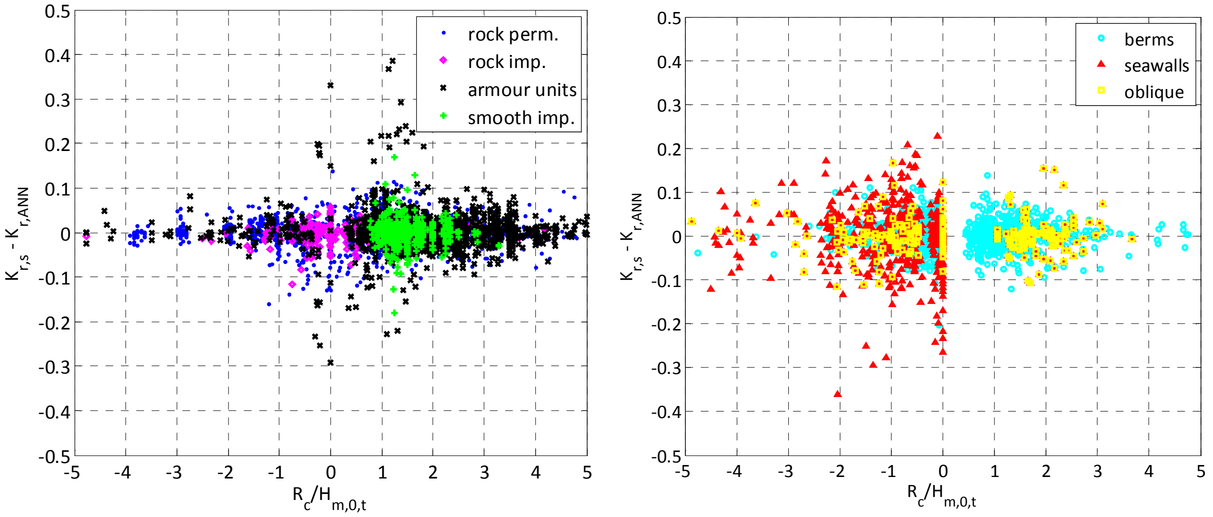


Figure 0.2 – *e* = $K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of the relative crest freeboard $R_c/H_{m,0,t}$ (abscissa); the data are distinguished and distributed into two plots according to the structure or wave attack type. Plot to the left: straight slopes cases; plot to the right: non-straight slopes and oblique wave attacks.

Then, Fig. 5.2 essentially provides two kind of information about the distribution of the error:

- the dependency on the relative crest freeboard (abscissa): from the diagrams it is pretty evident that the greater scatter is concentrated around the condition of zero-freeboard or low-emerged structures, while it sensibly decreases with the decrease of $R_c/H_{m,0,t}$ and is essentially absent for very submerged and emerged conditions.
- the dependency on the structure or wave attack type: as expected, the most scattered tests belong to the group of the seawalls (section F of the database), and, secondly, to the group of oblique wave attacks (section G). At the contrary, the groups of rock permeable and armour units (respectively, groups A and C) are associated to the lower errors, with the exception of a few dispersed points belonging to the group C.

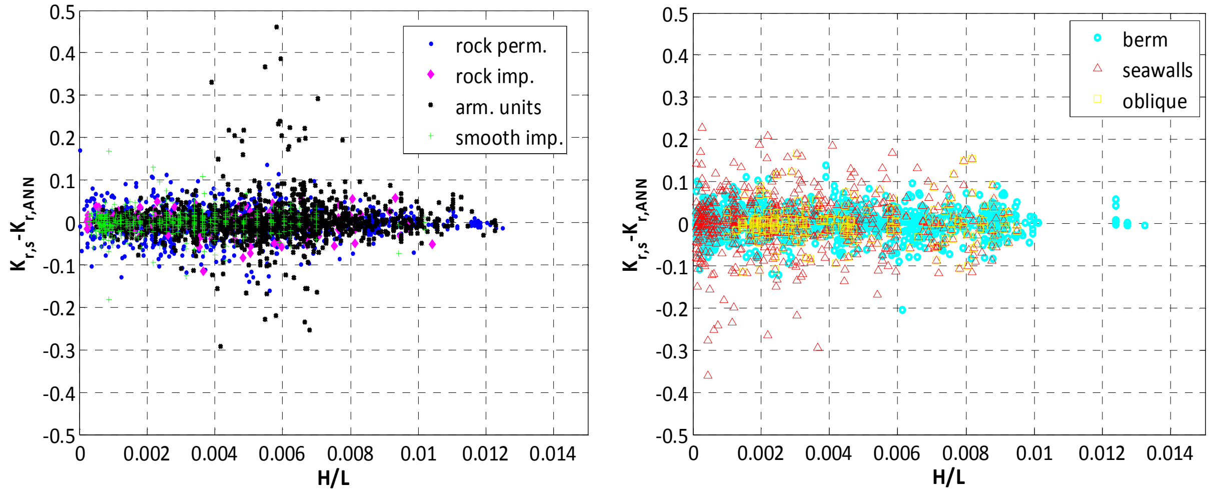


Figure 0.3 – *e* = $K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of the wave steepness $H_{m,0,t}/L_{m-1,0,t}$ (abscissa); the data are distinguished and distributed into two plots according to the structure or wave attack type. Plot to the left: straight slopes cases; plot to the right: non-straight slopes and oblique wave attacks.

Similarly, Fig. 5.3 aims to show the interlocking dependency of the error on the structure type and on the wave steepness. Actually, no particular influence of $H_{m,0,t}/L_{m-1,0,t}$ is detected. The non-straight slopes (groups E and F, plot to the right) and the cases of non-perpendicular wave attacks (group G, plot to the right) generally present more scatter, which modestly decreases at the increasing of $H_{m,0,t}/L_{m-1,0,t}$.

Concerning the straight slopes (groups A, B, C and D, plot to the left), the greatest scatter is associated to the few tests of group C and it is concentrated between $H_{m,0,t}/L_{m-1,0,t}$ = 0.004 and 0.008, i.e. the intermediate values of wave steepness.

The section C displays an anomalous behavior with respect to both $R_c/H_{m,0,t}$ and $H_{m,0,t}/L_{m-1,0,t}$, since most of its data is narrowly aligned around the condition of e = 0, while a small number of tests is extremely scattered. The analysis of the error distribution revealed that all the scattered points associated to the group C belong to the single dataset of Goda et al. (1975). The diagrams of Fig. 5.4 confirm this issue, showing the distribution of e for the group C as a function of respectively, $R_c/H_{m,0,t}$ (left) and $H_{m,0,t}/L_{m-1,0,t}$ (right) and enhancing the specific tests of Goda et al. (1975) with red colour.

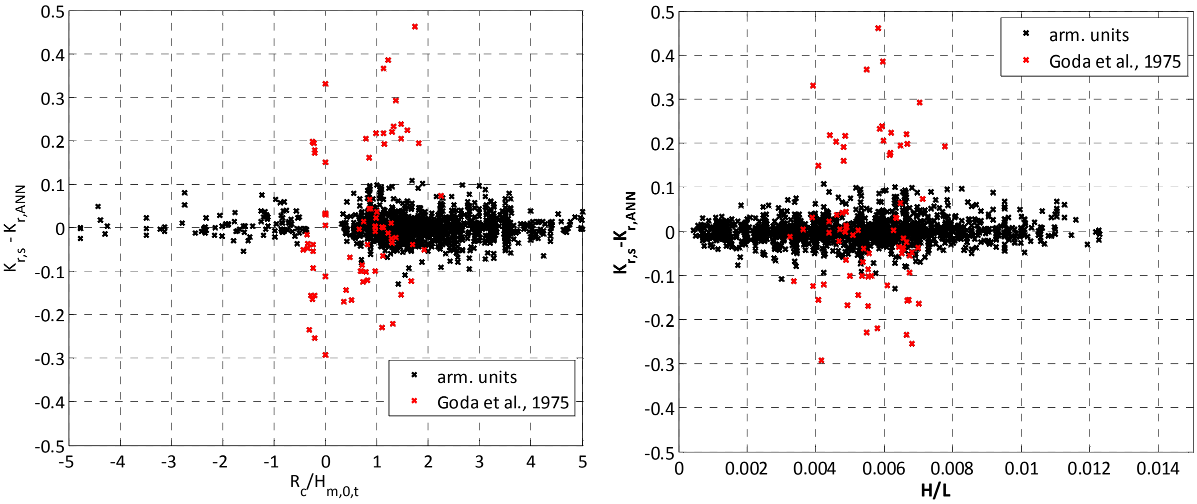


Figure 0.4 – $e = K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of the relative crest freeboard $R_c/H_{m,0,t}$ (abscissa, plot to the left) and of the wave steepness $H_{m,0,t}/L_{m-1,0,t}$ (abscissa, plot to the right). Tests belonging to the group C of "armour units"; the dataset of Goda et al., 1975, is enhanced with red colour.

The dataset of Goda et al. (1975) refers to a case of straight slopes covered by concrete tetrapods, included in the group C of "armour units". The plot of Fig. 5.5 aims to describe the distribution of the experimental values of K_r of this dataset in comparison to the predictions performed by Zanuttigh and van der Meer (2008) formula, as a function of the spectral breaker index $\xi_{0,p}$. This figure remarks the anomaly of the distribution of the experimental K_r of Goda et al. (1975), since no trend is detectable.

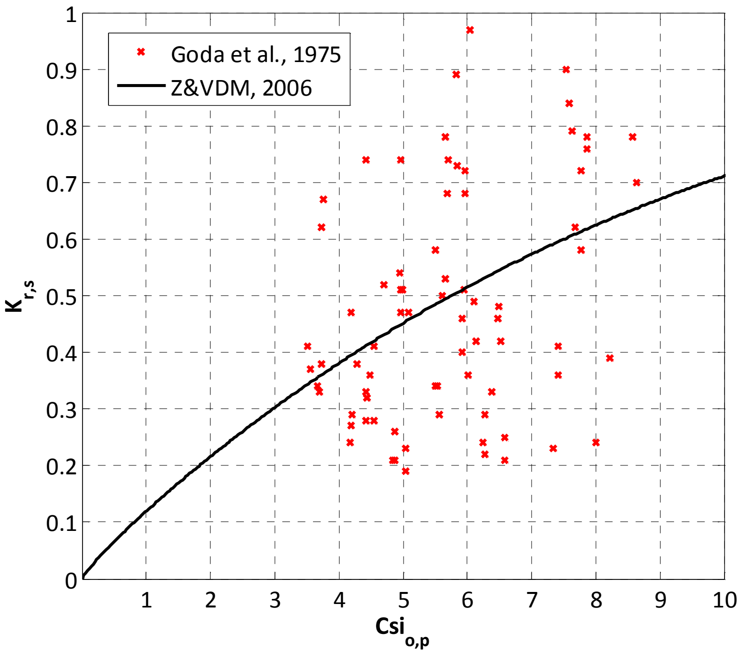


Figure 0.5 – $K_{r,s}$ **(ordinate) as a function of the spectral breaker index** $\xi_{0,p}$ **(abscissa). Tests belonging to the dataset of Goda et al., 1975 (red captions) compared to the estimates of** K_r **by Zanuttigh and van der Meer (2006) formula (continuous black line).**

The results of both Fig. 5.2 and Fig. 5.3 and the in-depth analysis allow concluding that the prediction of the ANN is not biased against the values of the main physical parameters, as no evident shortcoming is detected. On the contrary, the performance is instead affected by the belonging dataset and therefore by the complexity of the single tests.

By observing the distribution of the tests within the wave reflection database depicted in the pie-chart of Fig. 4.3, it is clear that there is a direct correspondence between number of available tests and performance of the ANN. Indeed, both groups F and G collect a poor percentage of data with respect to groups A and C which together include more than half of the data. Undoubtedly, the error is affected also by the higher degree of complexity of structures such as the seawalls or the oblique wave attacks.

Generally, it can be concluded that the contemporary effect of low number of available data and complex geometry determines a decrease of the ANN performance. This issue emphasizes the need to have more tests for complex cases and the wrong concept of low-weighting these cases through the definition of complexity and reliability factors, as already pointed out and discussed in Paragraph 4.4.4.

5.2.2 Wave transmission

The plots of Fig. 5.6 exactly represent the same outcomes of the ANN as in Fig. 5.1 for the wave reflection. By comparing these figures, it is evident how the ANN performance against the prediction of K_t is more accurate than against K_r, as it is enhanced by:

- the narrowness of the 95% error bands and the symmetry of the distribution of $K_{t,ANN}$ values around the ideal condition of the bisector line (plot to the left of Fig. 5.6);
- the very limited scatter and the little dispersion of the error $e = K_{t,s} - K_{t,ANN}$, which is almost all included between ±0.1 (plot to the left of Fig. 5.6). The highest errors ($e \approx 0.2 \div 0.4$) are relative to the lowest values of $K_{t,s}$, which – how the diagram confirms –

are approximately zeros. Therefore, these errors may be neglected, since the reliability of the corresponding measurements is almost null.

These qualitative diagrams reflect the excellence of the quantitative results reported in Tab. 5.1, where it can be appreciated that both *WI* and R^2 are extremely close to 1.

The plots of Fig. 5.7, which respectively represent the distribution of the error *e* as a function of the relative crest freeboard (to the left) and of the wave steepness (to the right), do not display any dependency of the scatter on these parameters. Furthermore, there is no evidence of shortcoming for none of the groups of structures, enhanced by the adoption of different captions in Fig. 5.7.

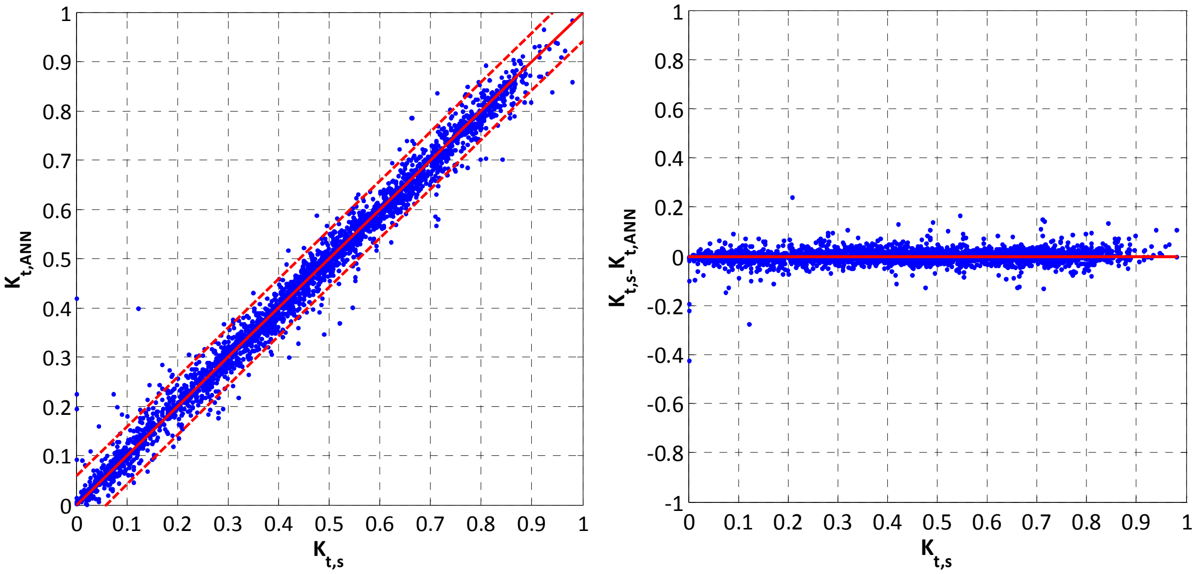


Figure 0.6 – Left: Comparison among K_t predicted values ($K_{t,ANN}$ ordinate) and corresponding K_t experimental values ($K_{t,s}$, abscissa); the continuous bisector represents the ideal condition ($K_{t,ANN} = K_{t,s}$), while the dashed lines refer to the 95% confidence levels. Right: difference $e = K_{t,s}$ - $K_{t,ANN}$ (ordinate) as a function of $K_{t,s}$ (abscissa). New advanced ANN.

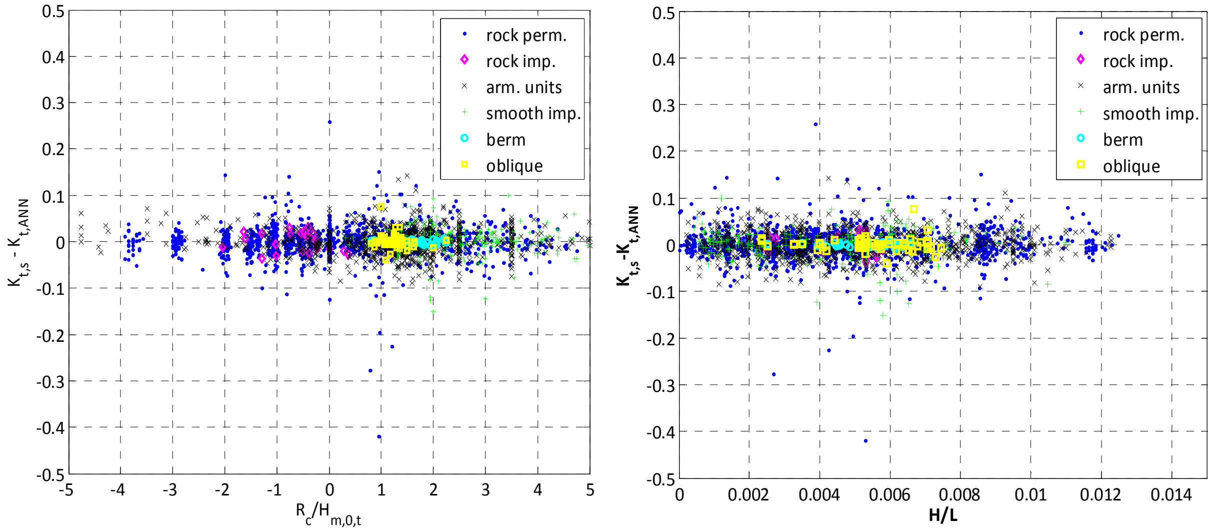


Figure 0.7 $e = K_{r,s}$ - $K_{r,ANN}$ (ordinate) as a function of the relative crest freeboard $R_c/H_{m,0,t}$ (abscissa, plot to the left) and of the wave steepness $H_{m,0,t}/L_{m-1,0,t}$ (abscissa, plot to the right); the data are distinguished and distributed into two plots according to the structure or wave attack type.

The comparison with existing ANNs (Paragraph 5.3.1.2) will demonstrate that this advanced ANN not only overcomes the problems of the work by Panizzo and Briganti (2007), but also improves the performance of the preliminary ANN for the wave reflection applied to the wave transmission (Formentin et al., 2013; see also Paragraph 4.5.2).

5.2.3 Wave overtopping

The results relating to the prediction of q are reported and discussed. It has to be noticed in advance that, following the work of Van Gent et al. (2007), all the experimental values of q $<10^{-6}$ $m^3/(sm)$ were discarded and associated to "zero-overtopping". With reference to the pie-chart of Figure 5.8 – which displays the distribution of q values, divided into different classes according to the order of magnitude – the loss of tests with overtopping $< 10^{-6}$ $m^3/(sm)$) corresponds to the 22% of the total amount of data.

The analysis of the effects of the implementation of a classifier ANN able to distinguish among "zero-overtopping" and "non- zero-overtopping", similarly to the studies of Verhaeghe et al. (2008) for the CLASH ANN, is investigated in the final Paragraph 5.5 of this chapter. Besides, an optimized criterion to deal with both of these classes of values is definitely proposed in the dedicated Chapter 6.

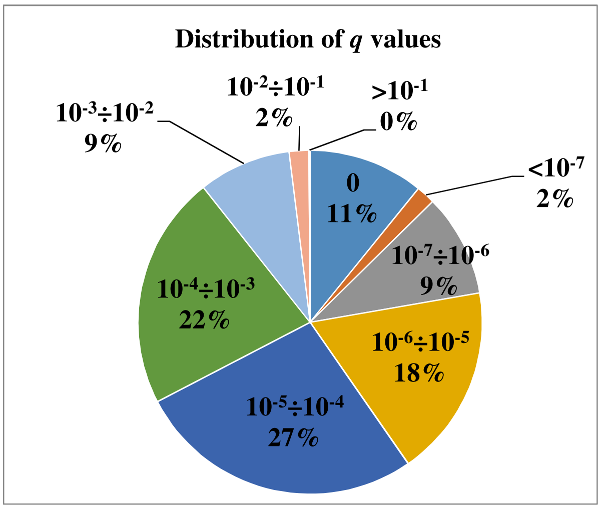


Figure 0.8 – Pie charts representing the distribution of the experimental values of q, divided into different classes according to the order of magnitude.

An overall presentation of the ANN performance is provided in Figure 5.9 through the qualitative comparison among the experimental (q_s) and the predicted (q_{ANN}) values of q (to the left) and through the distribution of the relative error $(q_s - q_{ANN})/q_s$ (to the right). These plots are similar to those of K_r and K_t (respectively, Fig.s 5.1 and 5.6), and the main differences are:

- the adoption of the logarithmic scale in the diagram (abscissa and ordinate, for the left plot and only abscissa for the right plot), to deal with the huge variability of the q values (6 orders of magnitude);
- the introduction of the relative error (right plot) in place of the absolute one; the relative has been preferred since it does not depend on the order of magnitude of the q, while the absolute one would decrease at the decrease of q.

In the diagram to the left, the bisector line still represents the ideal condition $q_{ANN} = q_s$ and the external lines are the 95% confidence bands. This plot shows a pretty good agreement among the predictions and the target values, in accordance with the numerical values of the error indexes (see Tab. 5.1).

The scatter is satisfactorily limited and most of the predictions fall within the confidence bands.

Despite the overall good performance, Fig 5.9-left shows that the distribution of the points is not completely symmetric, as ANN tends to systematically overestimate the low values of q ($q < 10^{-5}$ m^3/(sm)). Fig. 5.9 right clearly remarks this aspect: from this diagram, it is evident that the best ANN predictions are obtained for the higher values of q and that, generally, the lower q, the more pronounced the scatter and the asymmetry higher.

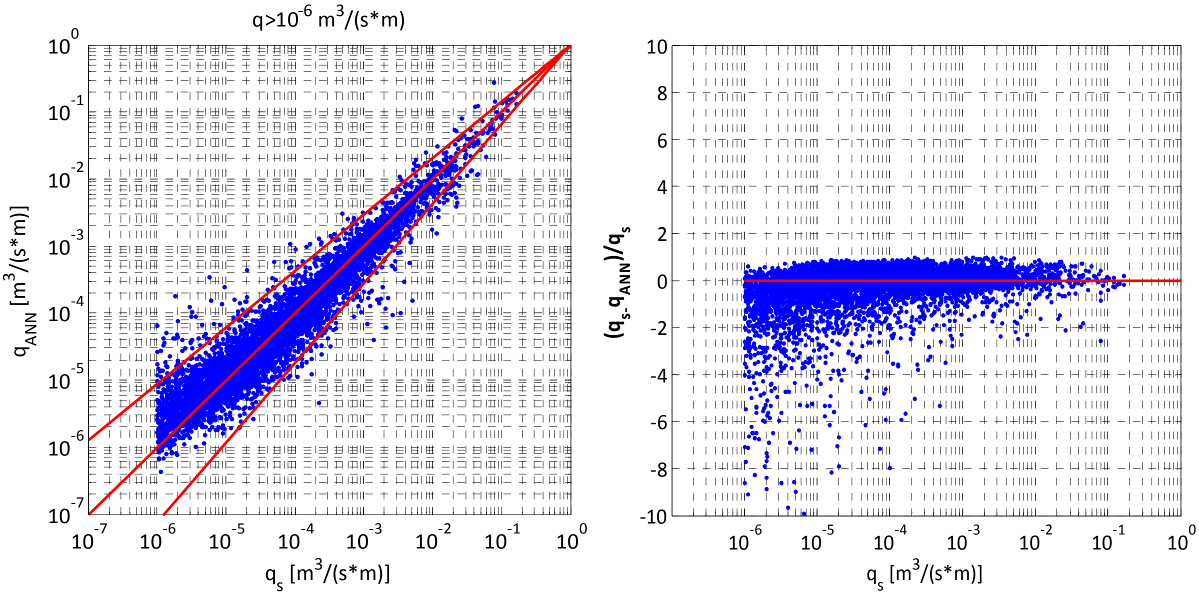


Figure 0.9 – Left: comparison among the predicted values of *q* (*q_{ANN}*, ordinate) and the corresponding experimental values (*q_s*, abscissa); the bisector represents the ideal condition (*$q_{ANN} = q_s$*), while the external lines refer to the 95% confidence levels. Predictions resulting from the average of 50 bootstrap resamples of the ANN. Right: Distribution of the relative error (*q_s-q_{ANN}*)/*q_s* (ordinate) as a function of the experimental values *q_s* (abscissa). The continuous line represents the ideal condition of 0-error.

One reason for the asymmetry of the error distribution may be attributed to the non-uniform distribution of the values of q_s within the complete interval [10^{-6}; 1] m^3/(sm). The 18% of values (nearly a fifth of the total amount of data) is between [10^{-6}; 10^{-5}] m^3/(sm) (see Fig. 5.8) and a 22% of the available tests are discarded as supposed to be too “small” and “un-reliable”. This means that the 40% of the complete database are values of q<10^{-5} m^3/(sm), yet more than half of this data are discarded.

The elimination of the “small” values, despite their numerical relevance, is expected to induce a bias of the prediction, i.e. to be the cause of the tendency to overestimate the low values of q_s. This assumption is confirmed by the results of the sensitivity analysis of the error distribution to the main physical parameters, i.e. the already presented $R_c/H_{m,0,t}$ and $H_{m,0,t}/L_{m-1,0,t}$.

Figure 5.10 displays the distribution of the relative error (q_s-q_{ANN})/q_s against the increasing values of $R_c/H_{m,0,t}$. The data are depicted into two separated diagrams in order to distinguish among straight slopes (panel to the left) and non-straight slopes and oblique wave attacks (panel to the right). It is interesting to note that, differently from K_r and K_t, in this case no test providing negative freeboard is included in the database.

Differently from Zanuttigh et al. (2014), Fig. 5.10 reveals that the greatest scatter is not concentrated around the condition of zero-freeboard, but it is generally spread on the complete range of $R_c/H_{m,0,t}$ values. This result indeed demonstrates that the ANN predictions are not biased with respect to $R_c/H_{m,0,t}$, the error is not correlated to this parameter. Moreover, from Fig. 5.10 it can be also concluded that there is not any dependency on the database section – i.e. on the type of structure (groups A, B, etc.) – since both the diagrams shows approximately the same level of scatter.

Fig. 5.11, which reports the distribution of $(q_s\text{-}q_{ANN})/q_s$ against the wave steepness, depicts a totally similar result: the ANN error is not affected neither by the values of $H_{m,0,t}/L_{m-1,0,t}$ nor by the belonging group of tests.

Therefore, the error analysis leads to the conclusion that the ANN performance is significantly dependent on the measure of the output parameter itself, q. The elimination of the “small” values of q not only limits the ANN field of validity, but also is responsible of a significant overestimation bias and affects the overall performance. The final solution to this matter is proposed and discussed within Chapter 6.

However, one important issue to be pointed out is that the bias of the prediction on the small values of q is always due to an overestimation error, which is definitely preferable for safe reasons.

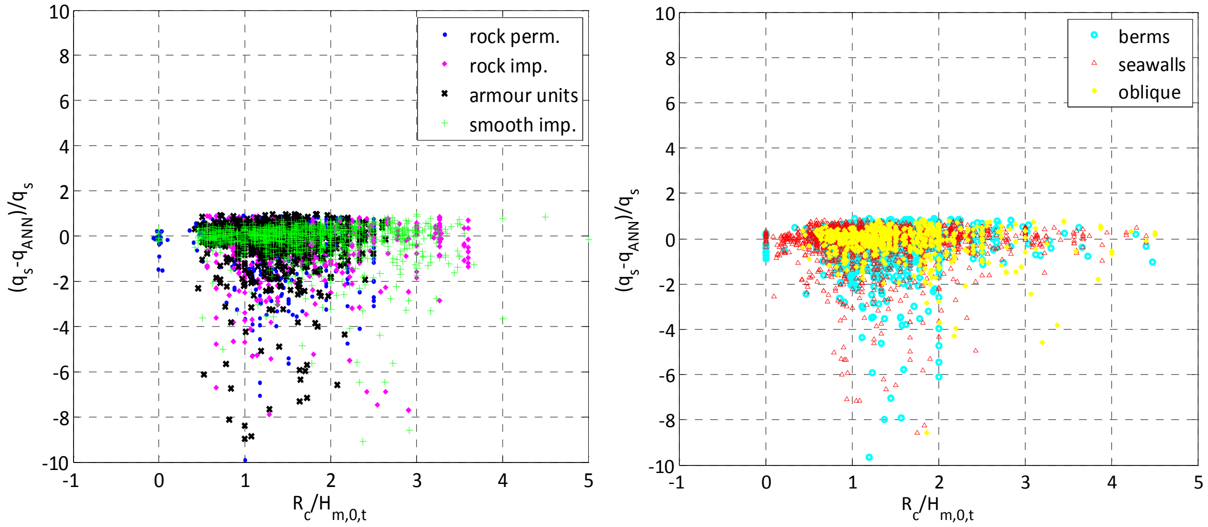


Figure 0.10 – Distribution of the relative error (*q_s-q_{ANN}*)/*q_s* (ordinate) as a function of the relative crest freeboard *R_c*/*$H_{m,0,t}$* (abscissa); the data are distinguished and distributed into two plots according to the structure or wave attack type. Plot to the left: straight slopes cases; plot to the right: non-straight slopes and oblique wave attacks.

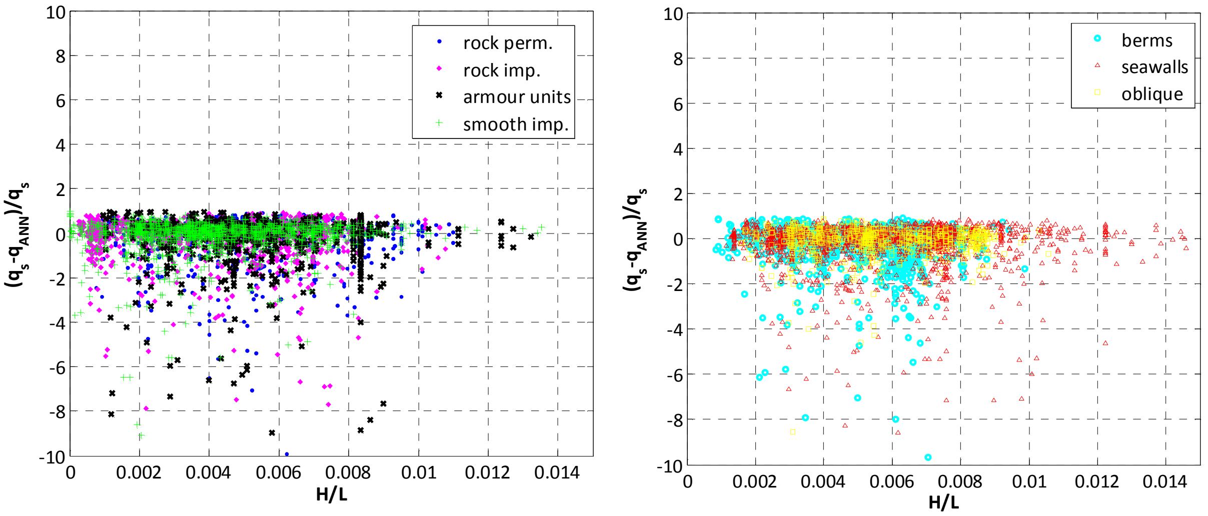


Figure 0.11 – Distribution of the relative error (*q_s-q_{ANN}*)/*q_s* (ordinate) as a function of the wave steepness *$H_{m,0,t}$*/*$L_{m-1,0,t}$* (abscissa); the data are distinguished and distributed into two plots according to the structure or wave attack type. Plot to the left: straight slopes cases; plot to the right: non-straight slopes and oblique wave attacks.

5.3 Comparison with existing tools

This section aims to compare the ANN performance with the available "tools" for the prediction of the three parameters K_r, K_t and q. In the first part (Paragraph 5.3.1) the comparison with the "traditional" tools, i.e. the literature formulae, will be investigated. The second part (Paragraph 5.3.2) is instead dedicated to present the comparison with the existing ANNs. In both cases, each process will be analyzed separately.

5.3.1 Comparison with existing formulae

The present Paragraph provides a comparison among the new ANN performance and the traditional prediction formulae for the three output parameters.

Due to limited ranges of validity associated to existing formulae, the comparison is carried out against narrower datasets selected among the whole databases employed to train the network.

5.3.1.1 Wave reflection

Similarly to the case of the preliminary ANN (see Paragraph 4.5.6) the ANN performance against the prediction of K_r is compared with the results of Zanuttigh and van der Meer (2008) (hereafter, ZVDM formula, see Eq. 4.9). According to the range and the field of validity of Eq. 4.9, only the straight slopes achieving the design conditions, have been selected, for a total amount of 1'219 tests (and, in details: 337 tests belonging to the group A, 117 to group B, 416 to group C and 349 to group D).

The ANN performance is qualitatively compared to the estimations obtained from ZVDM formula by means of two kind of diagrams, respectively reported in the Fig.s 5.12 and 5.13:

- the comparison between the distribution of the experimental K_r values and the ANN predicted ones around the trend of ZVDM formula (Fig. 5.12);
- the plots representing the dispersion of the K_r values predicted by either the ANN and ZVDM against the experimental K_r (Fig. 5.13).

The diagram to the left of Fig. 5.12 aims to display that the experimental K_r are approximately aligned along the ZVDM curves, revealing a satisfactory degree of reliability of the data themselves. However, some points do not precisely follow a specific trend against the breaker parameter $\xi_{0,p}$ (abscissa) and are "scattered" with respect to ZVDM lines.

Fig. 5.12 right remarks the ANN capability to reproduce not only the "aligned" experimental K_r, but also those "scattered" values that "escape" the formula trend. Therefore, the ANN demonstrates to be able to overcome the "limit" of the traditional formula which fails in the prediction of those values evidently depending on other parameters besides the ones individuated by the formula itself. This aspect is particularly evident when comparing in Fig. 5.12 the data belonging to the group of rock permeable and armour units (blue and black captions): the ANN essentially returns the same "scatter" (with respect to ZVDM curves) characterizing the experimental values. Differently, the groups of rock impermeable and completely impermeable slopes (purple and green captions) are less dispersed, being essentially distributed around the ZVDM curves.

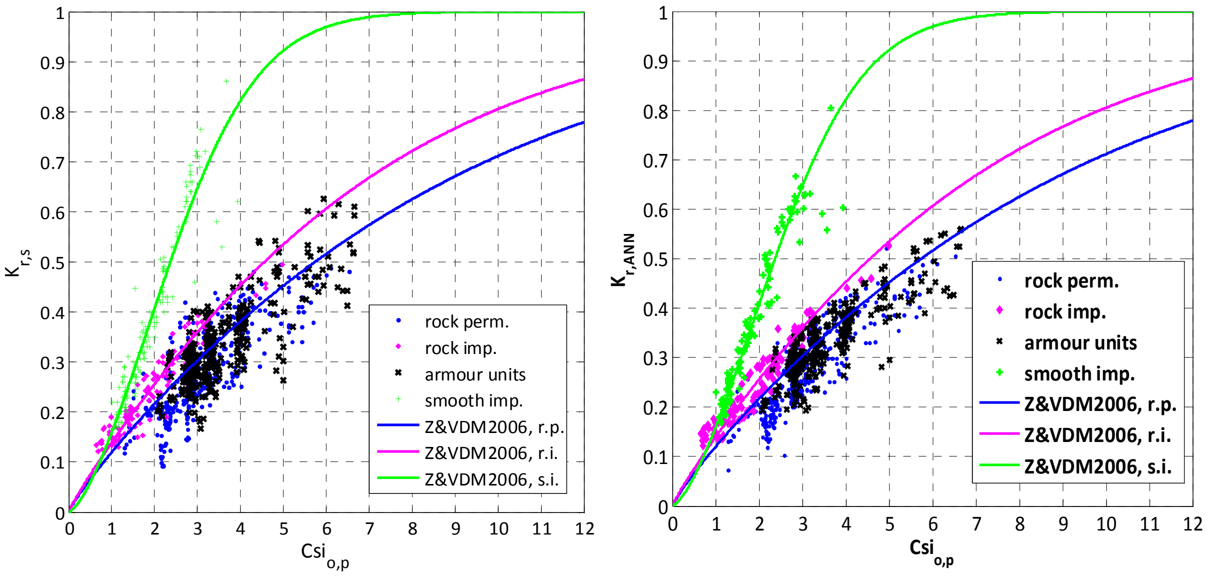


Figure 0.12 – Distribution of $K_{r,s}$ (ordinate, left panel) against the trend of ZVDM formula for the different structure types (continuous lines) compared to $K_{r,ANN}$ (ordinate, right panel) as functions of the breaker parameter $\xi_{0,p}$ (abscissa); the data are distinguished according to the structure type. Only straight slopes are included.

The plots of Fig. 5.13 and the numerical values of the *rmse* indexes computed for the QNN and the ZVDM formula for each of the four groups of structures (Tab. 5.2), provide a more direct comparison of performance is provided by. Fig. 5.13 shows that the values obtained from the ANN, $K_{r,ANN}$ (right panel) are characterized by a much lower dispersion around the ideal condition than the values $K_{r,ZVDM}$ (left panel). The sensibly narrower 95% confidence bands associated to the diagram of $K_{r,ANN}$ reflect the ANN lower values of *rmse* reported in Tab. 5.2. In this table, only the index *rmse* is displayed, in order to follow the work by Zanuttigh and van der Meer (2006) and provide a direct comparison with the indexes reported there.

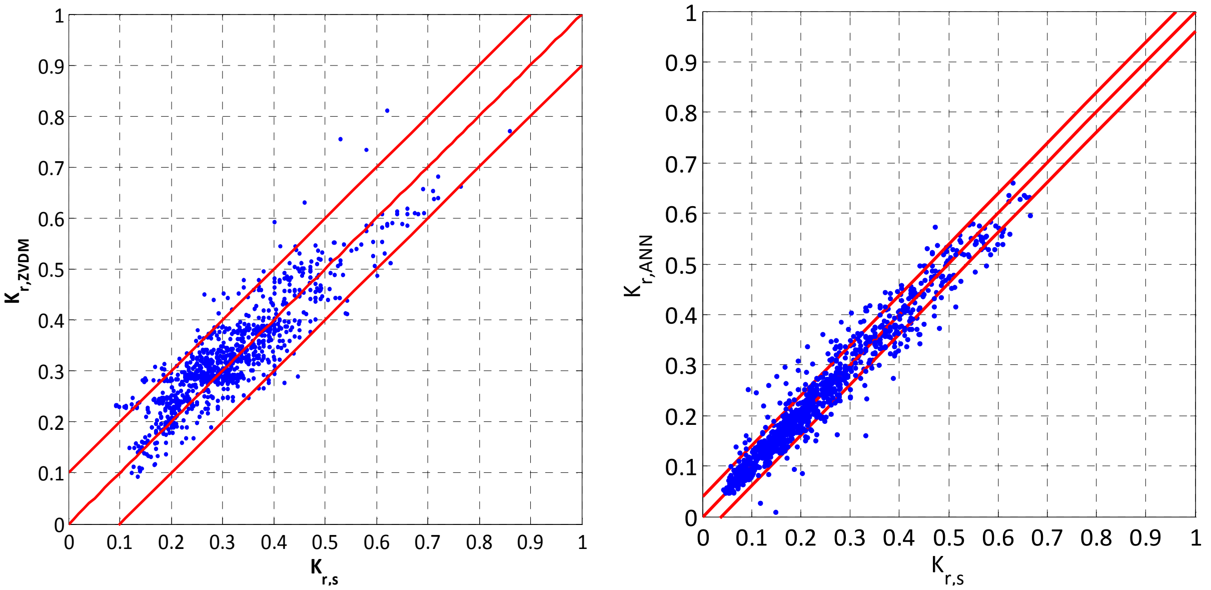


Figure 0.13 – Left: comparison among K_r values predicted values by ZVDM formula ($K_{r,ZVDM}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa). Right: comparison among K_r values predicted values by the new ANN ($K_{r,ANN}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa). The bisector represents the ideal condition ($K_{r,ZVDM}$ = $K_{r,s}$ or $K_{r,ANN}$ = $K_{r,s}$), while the external lines refer to the 95% confidence levels.

By comparing these results of the ANN to its overall performance (compare, e.g., the plot to right of Fig. 5.13 to the left plot of Fig. 5.1 left, or the *rmse* values of Tab. 5.2 to the average one relative to K_r in Tab 5.1), it is evident that the ANN behaves much better as long as it applied to a subset of tests in “design conditions”. This issue is a further confirmation of the main limit of the ANN, associated to the difficulty to represent very complex structures, as already pointed out in the Paragraph 5.2.1.

Table 0.2 – *rmse* values obtained simulating ANN and ZVDM formula against the straight slopes in “design conditions”. 4 groups of structures are individuated.

Type of structure (# nr. of tests)	*rmse* ZVDM	*rmse* ANN
Rock perm. (#337)	0.041	0.017
Rock imp. (#117)	0.038	0.005
Arm. Units (#416)	0.039	0.009
Smooth imp. (#349)	0.042	0.003

5.3.1.2 Wave transmission

The ANN performance is here compared to the formula by van der Meer et al. (2005) for the prediction of K_t (see Eq. 4.13, hereafter VDM formula). For practical applications, van der Meer et al. (2005) suggested to use the first relation of Eq. 4.13 for $G_c/H_{m,0,t} < 8$, the second one for $G_c/H_{m,0,t} > 12$ and to interpolate in the range $8 < G_c/H_{m,0,t} < 12$. Besides, some limits are imposed to VDM formula:

- $K_{t,l} = 0.05$, i.e. the 0.05 is the lower limit of validity;
- $K_{t,u} = -0.006\ G_c/H_{m,0,t} + 0.93$, i.e. a linear dependency for the relative crest is assumed as upper limit of validity;

Therefore, VDM formula is affected by a discontinuity in the range range $8 < G_c/H_{m,0,t} < 12$ and is upper and lower limited.

Table 5.2 shows the numerical results obtained applying the VDM formula and the ANN on the two datasets of the complete database separated in accordance to the two ranges of $G_c/H_{m,0,t}$. The ranges of validity of the formula are respected, as it is confirmed by the number of tests included in each of the two datasets whose sum is lower than the total number of 2’188 tests. Both the indexes *rmse* and R^2 are employed, with the purpose to provide a direct comparison with the values individuated in by van der Meer et al. (2005). From Tab. 5.2, the improvement carried out by the ANN with respect to VDM formula is evident. Considering the application of VDM, the different indexes reported in this work, with respect to van der Meer et al. (2005), are due to the extension of the database and in particular to the inclusion of smooth slopes.

Table 0.3 – *rmse* and R^2 values obtained simulating ANN and VDM formula against the wave transmission database.

Range **(# nr. of tests)**	**VDM formula**		**ANN**	
	rmse	R^2	*rmse*	R^2
$G_c/H_{m,0,t} < 8$ (#2'128)	0.109	0.71	0.030	0.98
$G_c/H_{m,0,t} > 12$ (#530)	0.082	0.81	0.028	0.98

The diagrams of Fig. 5.14 represent all the predictions performed by the formula (left panel) and the ANN (right) in comparison to the experimental values. When the formula is applied, the data were separately predicted and then re-jointed to be plotted together on the same diagram. Also from a qualitatively point of view, the performance of the new ANN is definitely better, both considering the very limited scatter, the narrowness of the confidence bands and the symmetry of the distribution of the points. Considering this latter aspect, a tendency to underestimate the low values and overestimate the high ones is detected for VDM formulae.

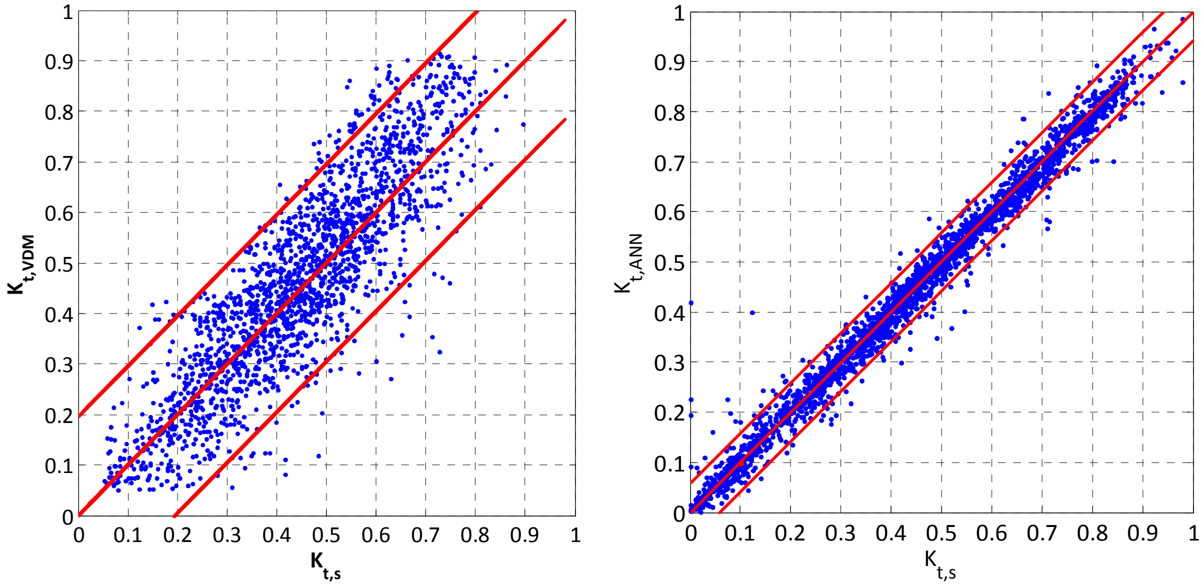


Figure 0.14 – Left: comparison among K_t values predicted values by VDM formula ($K_{t,VDM}$ ordinate) and corresponding K_t experimental values ($K_{t,s}$, abscissa). Right: comparison among K_t values predicted values by the new ANN ($K_{t,ANN}$ ordinate) and corresponding K_t experimental values ($K_{t,s}$, abscissa). The bisector represents the ideal condition ($K_{t,VDM} = K_{t,s}$ or $K_{t,ANN} = K_{t,s}$), while the external lines refer to the 95% confidence levels.

Analogous to the Paragraph 5.3.1.1 dedicated to the analysis of the wave reflection, the distribution of the ANN predictions around the curves of VDM formulae is evaluated by means of the qualitative diagrams of Fig. 5.15. This figure displays – for simplicity reasons - only the tests relative to a condition of zero freeboard ($R_c/H_{m,0,t} = 0$) and separates the data in different classes of the ratio $H_{m,0,t}/D = 0$. The two diagrams respectively report the experimental (left) and the ANN (right) predicted values of K_t as functions of the relative crest width $G_c/L_{m-1,0,t}$. This parameter – determining a discontinuity in VDM formulae – was selected in order to investigate the ANN capability to overcome the formulae shortcoming. In both the plots, two curves of VDM formulae are represented: they correspond to two "extreme" values of the breaker parameter $\xi_{0,p}$, i.e. $\xi_{0,p} = 1$ and $\xi_{0,p} = 10$. These values essentially correspond to the lowest and the highest experimental values present in the database, once applied the validity conditions imposed to the wave steepness by van der Meer et al. (2005).

By comparing the two diagrams of Fig. 5.15, two main issues are immediately detectable:

- the ANN predictions follow the distribution of the experimental values, even in case of a great "distance" between K_t and VDM curves or when a very low experimental K_t occurs. Therefore, no lower limit is detected for the new ANN, and this issue represents an improvement with respect to the preliminary ANN presented in the Paragraph 4.5.2.
- similarly to the case of K_r (see Paragraph 5.3.1.1), the ANN demonstrates to be able to deal with points which exceed the curves of the traditional formulae.

In conclusion, likewise the preliminary ANN, the new optimized ANN overcomes the fields of validity of van der Meer et al. (2005) formulae (see Eq. 4.13) and its discontinuity for $G_c/L_{m-1,0,t} = 0$, and is able to represent the dependence on D and $\xi_{0,p}$. The improvement can be explained by considering that the ANN adopts a greater number of parameters, achieving a more "complete" degree of interpretation of the wave transmission phenomenon.

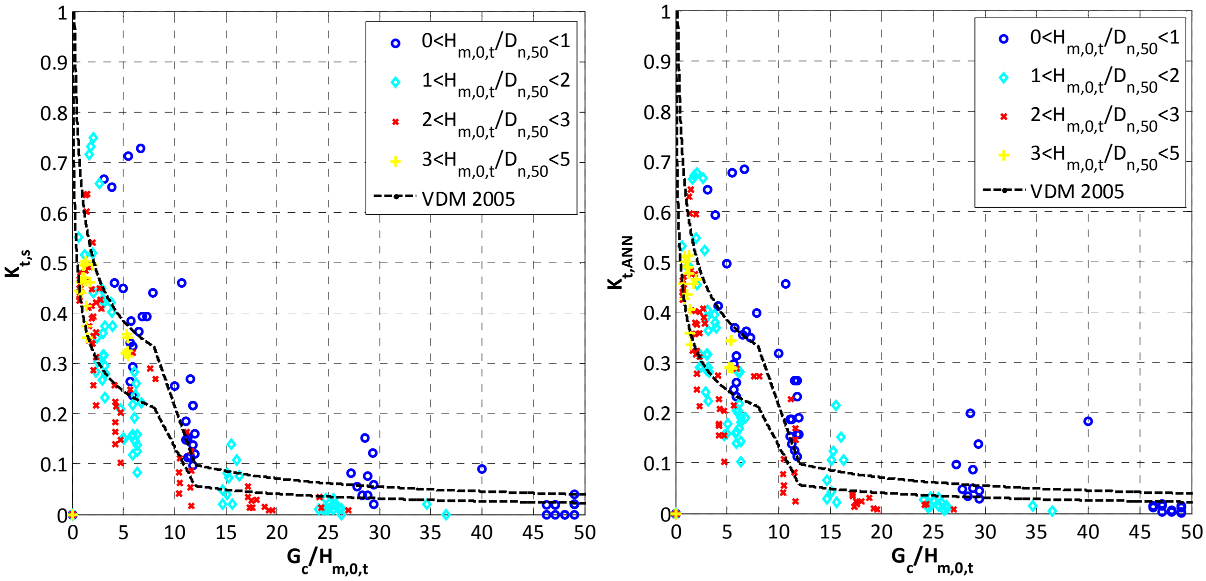


Figure 0.15 – Distribution of $K_{t,s}$ (ordinate, left panel) against the trend of VDM formula (continuous lines) compared to $K_{t,ANN}$ (ordinate, right panel) as functions of $G_c/L_{m-1,0,t}$ (abscissa); the data are distinguished according to the different classes of $H_{m,0,t}/D$. Only data characterized by $R_c/H_{m,0,t} = 0$ are included.

5.3.1.3 Wave overtopping

In presence of storm surge and waves the discharge is given both by the effects of the storm surge (overflow discharge) and on the incoming waves (overtopping discharge). In this case, it is indeed rather difficult to distinguish and accurately reproduce the contribution of wave overtopping and weir-like flow over the structure crest by means of theoretical approaches because of the discharge flowing back off-shore over the crests (Zanuttigh et al., 2008) in deeply submerged cases. For this reason, the following analysis will be focused on the cases of emerged and zero freeboard only. This way, it can be assumed that the total discharge rate be determined by the overtopping contribute only simplify the study.

EurOtop (2007) proposed a first approximation for the evaluation of the overtopping discharge, recently modified in van der Meer et al. (2013) to better approximate also the zero freeboard conditions. This formula, which is reported in Eq. 5.3 together with its upper limit, expressed by Eq. 5.4, will be hereafter referred as VDM.

$$q=\frac{0.026}{\sqrt{\tan\alpha}}\cdot\xi_{m-1,0}\cdot exp\left(-\left(2.7\frac{0.226}{\xi^{3}_{m-1,0}\cdot H_{m0}\cdot\gamma_b\cdot\gamma_f\cdot\gamma_v\cdot\gamma_\beta}\right)^{1.3}\right)\cdot\sqrt{gH^{3}_{m0}}$$

Eq. 0.3

with a maximum of

$$q_{max}=0.09\cdot exp\left(-\left(1.5\frac{R_c}{H_{m0}\cdot\gamma_f\cdot\gamma_\beta}\right)^{1.3}\right)\cdot\sqrt{gH^{3}_{m0}}$$

Eq. 0.4

It was not possible to trace the main field of validity of Eq.s 5.3 and 5.4. In van der Meer et al. (2013) it is said that the original EurOtop (2007) formulae were designed for "*sloping structures like dikes or levees* [...]. *Equation 2* [in EurOtop] *generally describes gentle slopes with plunging or breaking waves. In contrast, Equation 3 – the maximum overtopping – describes surging or non-breaking waves on fairly steep slopes* [...]. *Research in CLASH resulted in a lot of new data and in prediction formulae (Equations 2 and 3) for slopes, for breaking waves as well as non-breaking waves* [...]. *The final curve for non-breaking waves, covering the full range of relative freeboards is provided by Equation 7* [here Eq 5.3]. *The fit for breaking waves*, *Equation 6* [here Eq. 5.4], *is almost on top of the polynomial fit*."

Definitely, Eq.s 5.3 and 5.4 should reproduce the "*overtopping on sloping structures with zero and positive freeboard*", and Eq. 5.3 fits the non-breaking waves, while its maximum (Eq. 5.4) fits the breaking ones.

Because of the uncertainty connected to the actual fields of validity, and based on the assumption that "*sloping structures like dikes and levees*" should refer to the impermeable straights slopes, only the groups A, B, C and D of the database are included in this analysis and more attention is paid to the structures of group D (smooth impermeable). Both breaking and non-breaking waves, and zero-freeboard structures are included.

Following the methodology of Paragraphs 5.3.1.1 and 5.3.1.2, the ANN performance is qualitatively compared to VDM formulae, by means of two kind of plots:

- the distribution of the experimental and ANN predicted q values around the curve corresponding to VDM formulae (see Fig. 5.16). Due to the high number of parameters involved in Eq.s 5.3 and 5.4, the wave overtopping values are shown in the diagrams of Fig. 5.16 through the non-dimensional ratio $q/(gH^3_{m,0,t})^{0.5}$ (ordinate) as functions of the parameter $R_c/(H_{m,0,}t\cdot\gamma_f)$. Furthermore, the plotted curves correspond to the maximum of VDM (Eq. 5.4) and therefore have to be intended as an upper envelope. These choices allow to simplify the graphical display, avoiding to distinguish the data among different classes of $R_c/H_{m,0,}t$, $\xi_{0,p}$ and γ_f. The four diagrams of Fig. 5.16 display, from top to bottom, the data belonging to groups A and C (rock permeable and armour units, top panels) and to B and D (rock impermeable and smooth impermeable, bottom panels); from left to right, the experimental q (left panels) and the ANN predicted ones (right panels).
- the comparison among experimental and predicted q values derived by the two approaches (see Fig. 5.17); for similarity with Fig. 5.16, both the experimental q values (abscissa) and the predicted ones (ordinate) are still presented in Fig. 5.17 through the non-dimensional form of $q/(gH^3_{m,0,t})^{0.5}$. Differently from Fig. 5.16, Fig. 5.17 reports, for simplicity reasons, only the data referring to rock permeable and armoured slopes (groups A and C).

From the diagrams of Fig. 5.16, a rather great dispersion of the experimental values around the VDM curve is visible (left panels). This scatter – more pronounced for rock and smooth impermeable slopes than for permeable or amoured slopes (compare top panels to bottom ones) is pretty well represented by the ANN. Generally, despite the adoption of its upper limit, VDM formulae tends to provide a greater number of underestimation, especially for smooth slopes, than overestimation. This is partially unexpected, since VDM formula should have been fit against smooth dikes. It is evident that the VDM formula is here applied to several datasets that were not employed in the fitting and calibration process.

The direct comparison provided by Fig. 5.17, where VDM formulae are adopted in their "complete" form (i.e. the displayed values correspond to the minimum between Eq. 5.3 and 5.4), leads to the same conclusion already suggested by Fig. 5.16. The ANN is satisfactory able to reproduce a wider range of values of *q* than VDM formulae, which fields of validity should be pointed out (similarly to the wave reflection formulae by Zanuttigh and van der Meer, 2008 and the wave transmission ones by van der Meer et al., 2005).

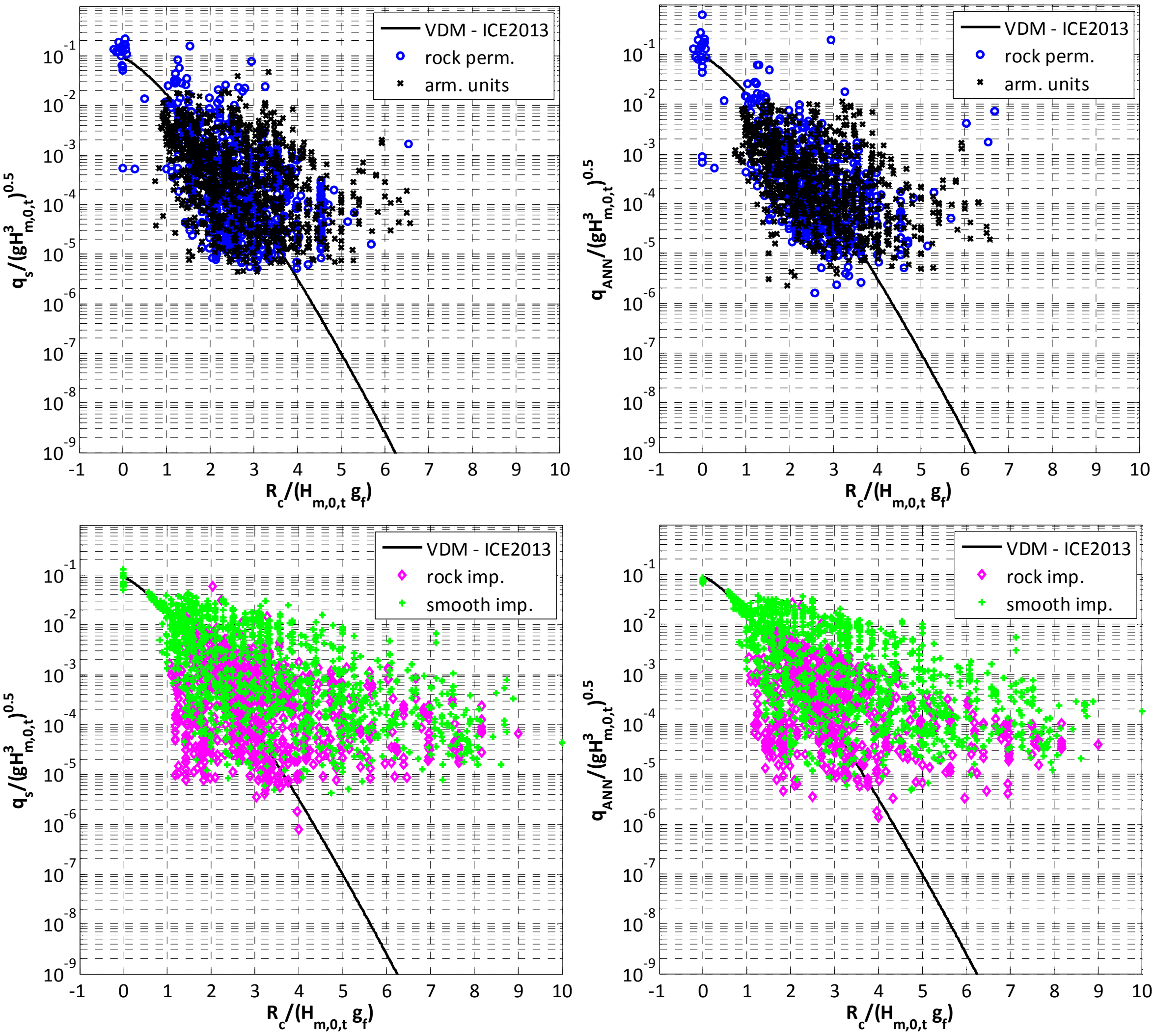


Figure 0.16 - Distribution of non-dimensional experimental q_s values (ordinate $q_s/(gH^3_{m,0,t})^{0.5}$, left panels) against the trend of VDM formula (continuous lines) compared to q$_{ANN}$ values (ordinate $q_{ANN}/(gH^3_{m,0,t})^{0.5}$, right panels) as functions of $R_c/(H_{m,0,t}\ \gamma_f)$ (abscissa). The data are distinguished according to the different armour type (top panels: rock permeable and armour units; bottom panels: rock impermeable and smooth impermeable slopes). Only data characterized by $R_c/H_{m,0,t} \geq 0$ are included.

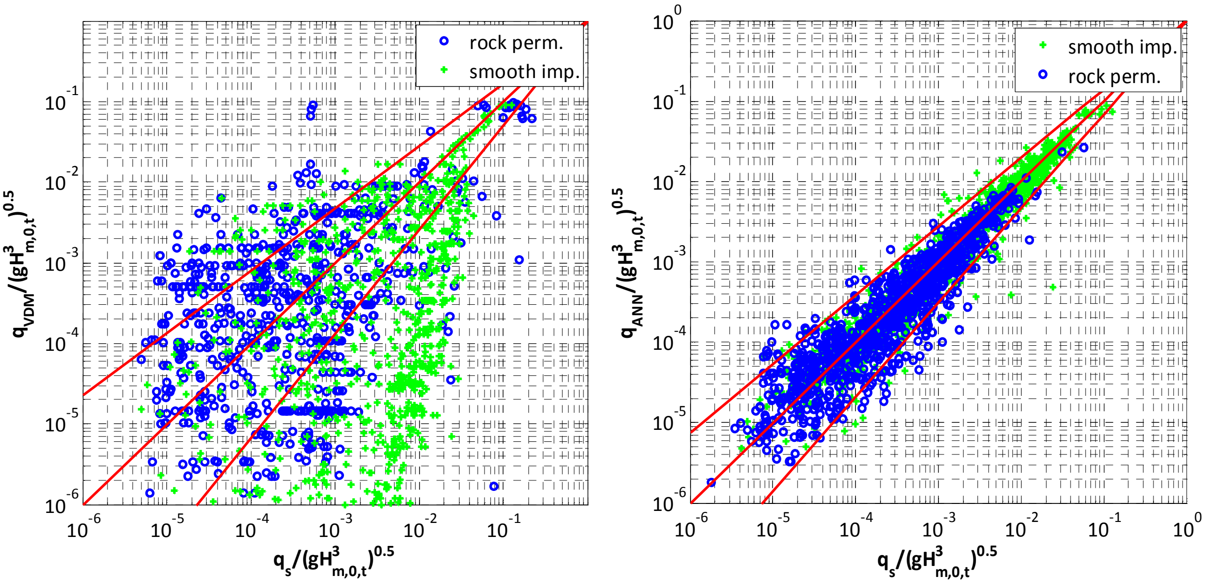


Figure 0.17 – Left: comparison among q values predicted by VDM formula ($q_{VDM}/(gH^3_{m,0,t})^{0.5}$, ordinate) and corresponding experimental ($q_s/(gH^3_{m,0,t})^{0.5}$, abscissa). Right: comparison among ANN predicted values ($q_{ANN}/(gH^3_{m,0,t})^{0.5}$, ordinate) and corresponding experimental ($q_s/(gH^3_{m,0,t})^{0.5}$, abscissa). The bisector represents the ideal condition ($q_{VDM} = q_s$ or $q_{ANN} = q_s$), while the external lines refer to the 95% confidence levels.

5.3.2 Comparison with existing ANNs

In order to complete the characterization of the new ANN skills and limits, a last analysis type of analysis is purposed. Since the new ANN is developed for the prediction of the three output parameters, it is important to prove that it behaves better than each of the existing ANNs which were instead optimized for the prediction of a single output. In other terms, it has to be demonstrated that the improvement carried out by the new ANN does not only consist in the realization of a single model, but it involves a better performance in the prediction of each of the outputs.

For this purpose, a comparison with the main existing ANNs is here presented and discussed. The comparison will not be limited to the analysis of the performance, but will be extended to the illustration of the differences between the ANNs, taking into account several aspects, such as:

- the models architecture;
- the number and the type of the input parameters;
- the characterization of the training database which strongly influences the ANNs fields of validity.

According to the analysis of literature presented in Chapter 3, the existing ANNs here considered are:

- the wave reflection ANN developed by Zanuttigh et al. (2013), see Paragraph 3.5;
- the wave transmission ANN developed by Panizzo and Briganti (2007), see Paragraph 3.4;
- the wave overtopping ANN developed within the CLASH context (Van Gent et al., 2007), see Paragraph 3.3.

The input parameters characterizing the three ANNs are synthesized in Tab. 5.4, where the CLASH symbolism is employed. The following subsections 5.3.2.1, 5.3.2.2 and 5.3.2.3 are dedicated to the specific analysis of each of the abovementioned ANNs.

Table 0.4 – Input parameters of the three existing ANN considered in this work: the ANN by Zanuttigh et al. (2013) for K_r; the ANN by Panizzo and Briganti (2007) for K_t; the CLASH ANN by Van Gent et al., (2007) for q.

#	K_r (Zanuttigh et al., 2013)	K_t (Panizzo and Briganti, 2007)	q (Van Gent et al., 2007)
1	$H_{m,0,t}/L_{m-1,0,t}$	$R_c/H_{m,0,t}$	$H_{m,0,t}$
2	$h_t/L_{m-1,0,t}$	$G_c/L_{m-1,0,t}$	$T_{m-1,t}$
3	γ_f	$G_c/H_{m,0,t}$	γ_f
4	$cot\alpha_d$	$H_{m,0,t}/h$	$cot\alpha_d$
5	$cot\alpha_{incl}$	$\xi_{0,p}$	$cot\alpha_u$
6	$D/H_{m,0,t}$	$H_{m,0,t}/D$	B
7	$R_c/H_{m,0,t}$		B_t
8	$B/L_{m-1,0,t}$		h
9	$h_b/H_{m,0,t}$		h_t
10	$G_c/L_{m-1,0,t}$		h_b
11	m		R_c
12	β		A_c
13	*Spreading*		G_c
14			$tan\alpha_B$
15			β

5.3.2.1 Wave reflection

Since the ANN by Zanuttigh et al. (2013) has been already presented in this work (chapter 4), it is worthy to stress some paramount issues before proceeding in the analysis of performance:

- differently from Paragraph 4.3, where a mere comparison among input parameters were purposed, in the present context the existing wave reflection ANN is considered in its original and complete form – including the training database.
- The "preliminary" ANN presented in section 4.5.1 actually corresponds to the ANN by Zanuttigh et al. 2013, and it was there considered in its original features and applied to the original database. The performance of that ANN is hereafter purposed again (see Tab. 5.5 and Fig. 5.18) in comparison to the new advanced ANN in order to show the achieved improvement.

The new ANN was definitely optimized based on the existing ANN by Zanuttigh et al. (2013). The architecture, the number and the type of input parameters are indeed very similar, but some essential differences should be pointed out:

- the training database, which has been extended from 5'871 data to 7'413 (see Paragraph 4.2.2);
- the elimination of some input parameters (such as $h_t/L_{m-1,0,t}$ and the *spreading*) and the introduction of some new ones (e.g., $B_t/L_{m-1,0,t}$, $h/H_{m,0,t}$, etc.); a complete and detailed comparison is reported in Tab. 4.4, where the process of optimization of the new ANN input set is discussed.
- the adoption of the bootstrapping resample technique instead of the early stopping to improve and assess the ANN capability of generalization.

Keeping in mind the previous considerations, the quantitative and qualitative performance comparisons can be developed. Tab. 5.5 provides the comparison among the error indexes, while Fig. 5.18 displays the distribution of the ANNs predictions against the experimental values. From Tab. 5.5, the advances achieved by the new ANN can be detected in terms of increasing of both *WI* and R^2. These improvements are partially balanced by the increase of the standard deviations – which is particularly evident for R^2 – and by the number of large errors. This can be explained by the enlargement of the database which essentially regarded complex cases of reshaping berm breakwaters and seawalls (see Paragraph 4.2.2).

Table 0.5 – Comparison among the quantitative performance of the new ANN and the existing ANN developed by Zanuttigh et al. (2013) for the prediction of the wave reflection coefficient.

Prediction of the wave reflection coefficient, K_r				
ANN	*RMSE*	*WI*	R^2	# large errors (%)
existing ANN	0.038 ± 0.003	0.985 ± 0.003	0.943 ± 0.006	2.5%
new ANN	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%

The diagrams of Fig. 5.18, where the performance of the “old” ANN is displayed in black and white colours, respect the numerical results. The inclusion of new data (most of which are concentrated in the upper part of the right panel, corresponding to the new ANN) denotes an increased availability of tests with high values of K_r, which reinforce and extend the new ANN field of validity, and reduce its tendency to underestimate such values (indeed, the panel to the left, corresponding to the “old” ANN, shows a majority of points under the bisector in the upper part). However a little greater scatter is presented by the new ANN, in correspondence of $K_{r,s}$ = 0.7÷0.8, which is likely to be induced by the specific dataset of Goda et al. (1975), see Paragraph 5.2.1.

In conclusion, despite the little increased scatter, the new ANN undoubtedly achieves a better performance, enlightened by the very higher values of *WI* and R^2 and by the extension of the validity field.

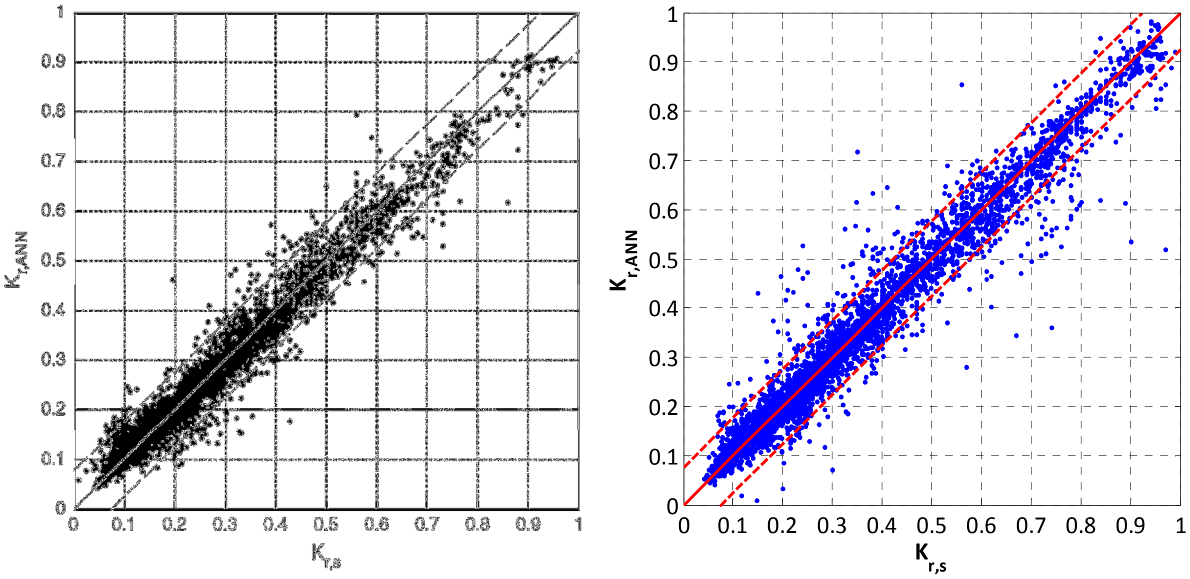


Figure 0.18 - Left: comparison among K_r predicted values by the existing ANN by Zanuttigh et al. 2013 (ordinate) and corresponding K_r experimental values (abscissa). Right: comparison among K_r predicted values by the new ANN (ordinate) and corresponding K_r experimental values (abscissa). The bisector represents the ideal condition ($K_{r,ANN}$ = $K_{r,s}$), while the external lines refer to the 95% confidence levels.

5.3.2.2 Wave transmission

The ANN by Panizzo and Briganti (2007) adopted a sensible lower number of input parameters than the new ANN. From Tab. 5.4, it can be observed that the parameters composing the input set of the “old” ANN for K_t are essentially finalized to the description of the wave attack conditions, rather than the cross-section geometry. This was probably due to the employed database, concerning exclusively low crested structures. The only “structural” quantities are G_c and R_c, for the characterization of the crests.

The new ANN has in common with this existing ANN an important feature: the adoption of non-dimensional parameters. As already proved (see Paragraph 4.3 and the sensitivity analysis to the input parameters), this issue leads to a good improvement, since it permits a more physically based scaling (instead of a scaling based on similarity laws).

The training database of the existing transmission ANN included 2’285 tests derived from the DELOS project (van der Meer et al., 2005) only account. Some data were excluded from the database following the criterions explained in d'Angremond et al. (1996). Being the new ANN trained on 3’366 tests, the adoption of different databases is, for this application, rather significant (the new ANN has nearly 50% additional data).

Concerning the ANN architecture, the only available information regarding the existing ANN are the adoption of a sigmoid transfer function for the computation of the hidden neurons (similarly to the new ANN) and the employment of 6 hidden neurons in a single hidden layer. No specific reference is made to the training algorithm, which is supposed to belong the category of gradient descent algorithms (see Paragraph 2.3.5). No mention is done to the bootstrap resampling technique.

Table 0.6 – Comparison among the quantitative performance of the new ANN and the existing ANN developed by Panizzo and Briganti (2007) for the prediction of the wave transmission coefficient.

Prediction of the wave transmission coefficient, K_t				
ANN (# nr. of tests)	*RMSE*	*WI*	R^2	# large errors (%)
existing ANN	0.065	-	0.983	-
new ANN	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%

In similarity to the application of the wave reflection, the comparison between the two ANNs is provided through the numerical indexes (Tab. 5.6) and the qualitative diagrams of Fig. 5.19. For this application, the values of *WI* and the standard deviations associated to the indexes were not available for the existing ANN. The numerical comparison can be therefore provided in terms of *rmse* and R^2: the first index indicates a greater dispersion of the predictions and a larger confidence interval for the ANN by Panizzo and Briganti, while the second one, which is almost equal between the two ANNs, reveals a similar degree of correlation between predictions and measurements. The qualitative comparison leads to the conclusion that the new ANN behaves better, being the predictions evidently more accurate and less scattered.

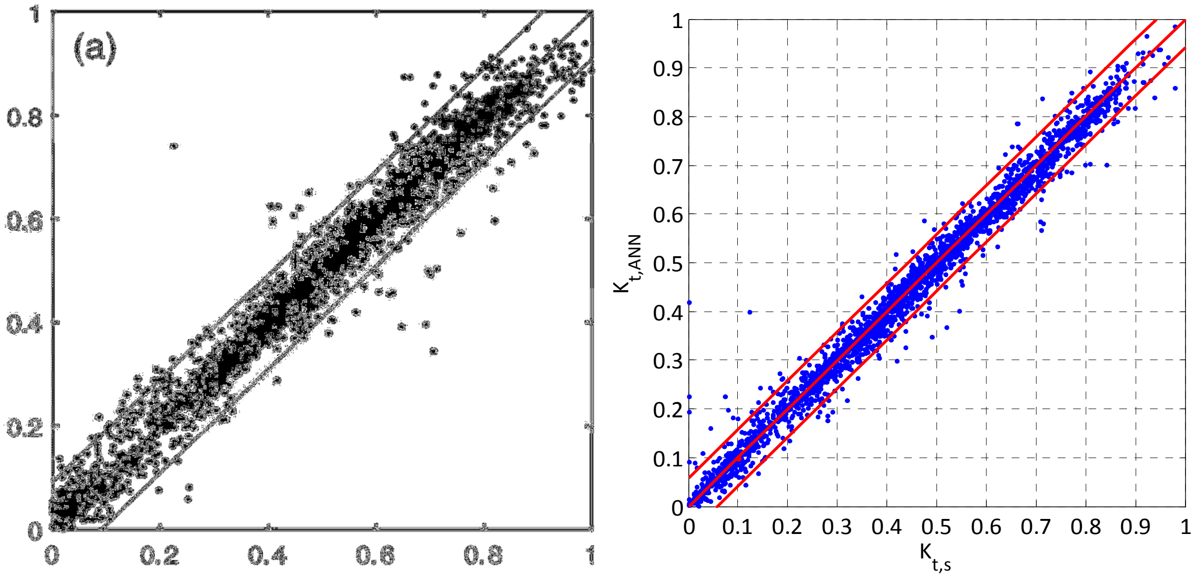


Figure 0.19 – Left: comparison among K_t predicted values by the existing ANN by Panizzo and Briganti, 2007 (ordinate) and corresponding K_t experimental values (abscissa). Right: comparison among K_t predicted values by the new ANN (ordinate) and corresponding K_t experimental values (abscissa). The bisector represents the ideal condition ($K_{t,ANN}$ = $K_{t,s}$), while the external lines refer to the 95% confidence levels.

5.3.2.3 Wave overtopping

The new ANN and the CLASH ANN (Van Gent et al., 2007) have in common nearly the same number and type of input parameters (see Tab. 5.4), but the paramount difference is the scaling process. The CLASH ANN adopts dimensional parameters (e.g., $T_{m-1,0,t}$, R_c, etc.) but scaled quantities with respect to $H_{m,0,t}$ = 1 (for further details, see Paragraph 3.3), while the new ANN adopts non-dimensional parameters through a physically based scaling.

Other relevant features characterizing the new ANN are:

- the elimination of the early stopping, which instead was adopted by Van Gent et al. (2007);
- the transformation of the output q into q^* as explained in Eq.s 5.1 and 5.2, in order to provide a narrow-varying target set to the ANN;
- the extension of the overtopping database with the inclusion of new data on steep smooth slopes and seawalls (see Paragraph 4.2.1);
- the choice to perform the bootstrap resampling of the database without the weight factors (please, note that for Van Gent et al. (2007) the bootstrapping was employed for the assessment of the uncertainty, and not as an improving generalization technique);
- the number of hidden neurons, which is 40 instead of 20;

Because of the different way of defining the target values (q^* for the new ANN instead of log (q'), where q' is the scaled correspondent to q in Froude similarity law, for the CLASH ANN), the *rmse* indexes of the two ANNs cannot be directly compared (see Paragraph 5.2 and Zanuttigh et al., 2014). Since Van Gent et al. (2007) did not point out any other error index, the quantitative comparison is not possible. A preliminary ANN for the wave overtopping – which adopted the same target log (q') = log ($q/(gH^3_{m,0,t})^{0.5}$) – was presented in Zanuttigh et al. (2014). That preliminary ANN differed from the new definitive one only in the way the targets were scaled and presented to the ANN. This aspect indeed affects the ANN performance (as proved in Zanuttigh et al., 2014), however the *rmse* associated to that preliminary

ANN is here reported (Tab. 5.7) in order to purpose an approximate numerical comparison with the CLASH ANN.

It is worthy to remark that it is not possible to convert the *rmse* value associated to q^* into the scale of the *rmse* associated to $\log(q/(gH^3_{m,0,t})^{0.5})$ (or vice versa) since the argument *X* of the function *rmse* (Eq. 4.3) is different for the two approaches (see, respectively, Eq. 5.5 and 5.6) and the transformation cannot be reversed.

$$X_{CLASH}=log\frac{q}{\sqrt{gH^3_{m,0,t}}}$$

Eq. 0.5

$$X_{NEW}=q^*=\frac{log_{10}(q_{AD})-min\{log_{10}(q_{AD})\}}{|\ min\{log_{10}(q_{AD})\}-max\{log_{10}(q_{AD})\}|},\quad q_{AD}=\frac{q}{\sqrt{gH^3_{m,0,t}}}$$

Eq. 0.6

In Tab. 5.7, the value of *rmse* = 0.28 associated to the preliminary ANN is lower than the CLASH ANN *rmse* = 0.29. This result is supposed to be further improved by the new optimized ANN, as it can be deduced by comparing the values of *WI* and R^2 between the preliminary and the optimized ANN.

Table 0.7 – Comparison among the quantitative performance of: the existing ANN developed by Van Gent et al. (2007); a preliminary ANN for the prediction of the wave overtopping discharge, which differs from the definitive new ANN in the way the target values are presented to the network; the new definitive ANN.

Prediction of the wave overtopping discharge, $q>10^{-6}$ m^3/(sm)					
ANN	Target	*RMSE*	*WI*	R^2	# large errors (%)
existing ANN	$\log(q/(gH^3_{m,0,t})^{0.5})$	0.29	-	-	-
preliminary ANN	$\log(q/(gH^3_{m,0,t})^{0.5})$	0.28 ± 0.02	0.976 ± 0.005	0.91 ± 0.01	0.3%
new ANN	q^*	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%

The qualitative comparison is provided by the plots of Fig. 5.20. Also in this case the comparison was not straightforward, since for Van Gent et al. (2007) the bands refer to the 90% confidence levels, while in this work the new ANN has been generally presented with the 95% confidence intervals. However, in this case the computation of the 90% confidence intervals was of course possible and the diagram correctly represented.

From Fig. 5.20 a similar degree of scatter is detected. Though, the new ANN predictions (panel to the right) appear to be more biased for small values of *q* (in the range $q<10^{-5}$ m^3/(sm)).

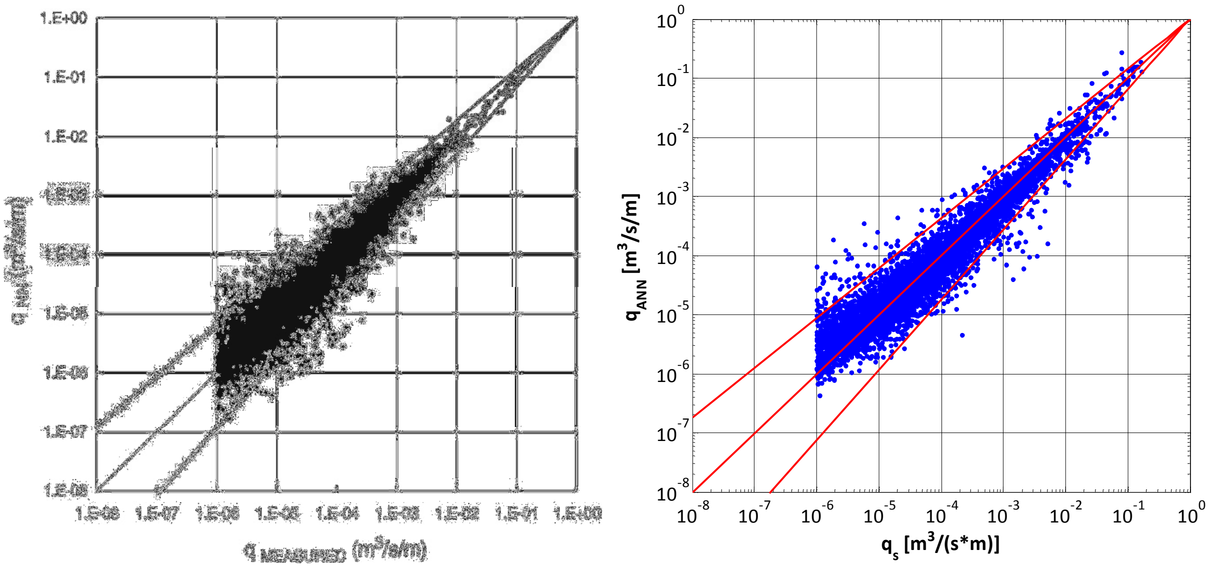


Figure 0.20 – Left: comparison among *q* predicted values by the existing ANN by Van Gent et al., 2007 (ordinate) and corresponding *q* experimental values (abscissa). Right: comparison among *q* predicted values by the new ANN (ordinate) and corresponding *q* experimental values (abscissa). The bisector represents the ideal condition (q_{ANN} = q_s), while the external lines refer to the 90% confidence levels.

In conclusion, at the present state, the new ANN predictions are approximately as accurate as the CLASH ANN ones, despite a reduced quantitative error is detected. The scarce improvement may be ascribed to the extension of the database and the inclusion of "complex" structures. For this reason, an in-depth analysis aimed to assess the new ANN capability to deal with complicate geometry (berms and toes) and out-of-range values (obtained through an artificial extension of the database, by varying the toe and berms dimensions) is needed and will constitute the basis of a further research. Furthermore, Chapter 6 is completely dedicated to the development of a technique to solve (or reduce) the overestimation failing of the new ANN.

5.4 Contemporary predictions

The aim of this section is to verify the ANN behavior when applied to a "contemporary" prediction and definitely to answer the following question: does the ANN performance improve when more output parameters are contemporary read as targets? The question is reasonable, standing the physical correlation among the three output parameters, and the consequent guess that a contemporary information could enforce the ANN ability to learn the input-output patterns.

A "contemporary" prediction essentially involves a modification of the ANN architecture to include multiple neurons in the output layer and a re-definition of the training database in order to individuate the tests for which all the output parameters are available. Fig. 5.21 schematically represents the conceptual layout of such modified architecture: the number of the connection arises, since each of the output neurons is connected to each of the hidden neurons. This issue leads to the consideration that the minimum number of tests to train the ANN properly arises as well (see Paragraph 4.3). According to the relationship reported in Eq. 4.2., in case of two output neurons (e.g., K_r and K_t) the minimum required number becomes 844 and for three output neurons (K_r, K_t and q) it becomes 886 (instead of 802, see Tab. 4.3).

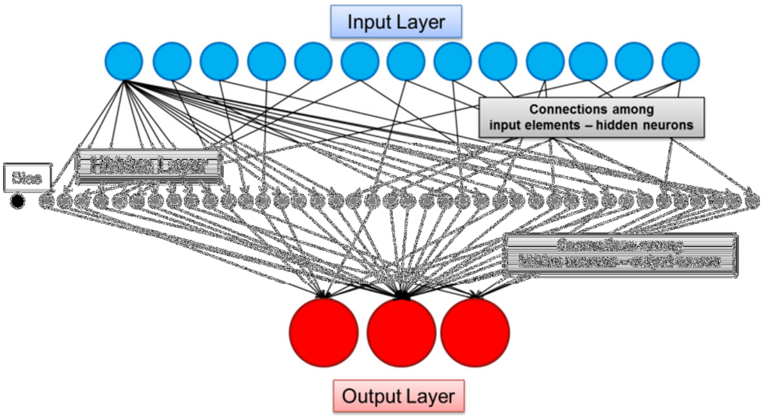


Figure 0.21 - Schematization of the conceptual layout of the modified architecture of the ANN for the inclusion of more than one output neurons.

The matter of the minimum number of available tests is expected to strongly affect the actual feasibility of a “contemporary” prediction. Indeed, as anticipated in the Paragraph 4.2.4, the common “databases” are far smaller than the single datasets of wave reflection, transmission and overtopping. In particular, the percentage of tests for which both q and K_t are known is insignificant (less than 1%), and for this reason the attempt to train an ANN with these two outputs is *a priori* excluded.

The contemporary predictions of K_r and K_t and of K_r and q were instead investigated, and the derived results are presented and discussed in the next two sections (respectively, Paragraph 5.4.1 and 5.4.2). Then, a comparison with the “single” original predictions is proposed and some conclusions are drawn in the conclusive Paragraph 5.4.3.

5.4.1 Contemporary predictions of K_r and K_t

In this section, the performance of the ANNs with the respect to the contemporary inclusion of both the output parameters K_r and K_t is investigated. The training database has been reduced to 2’303 tests at all (the 31% of the total amount of tests available for wave reflection, i.e. 7’413 tests, and the 68% of the total amount of tests available for wave transmission, i.e. 3’366 tests). The pie chart of Fig. 5.22 illustrates the reduced database and shows the distribution of the residual available tests: nearly the totality of data refers to rock permeable or armoured straight slopes. The inequality of the distribution seriously affects the ANN applicability and field of validity even more than the shortage of tests. It is important to keep in mind this aspect, when comparing the numerical or graphical results of the contemporary predictions to the single ones.

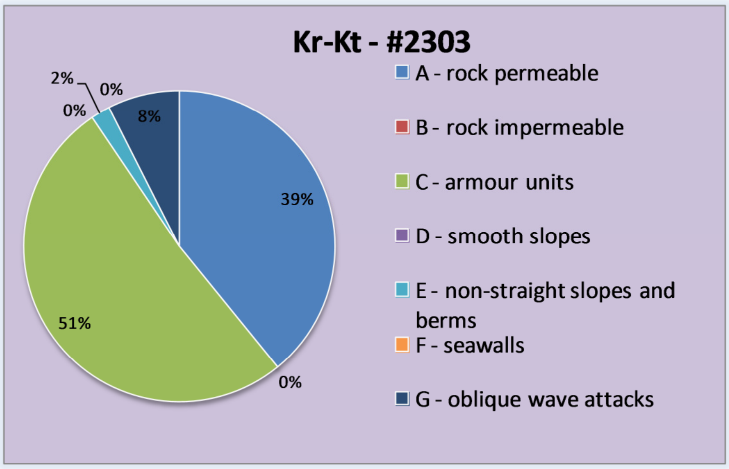


Figure 0.22 – Pie chart representing the distribution of the tests containing both the wave reflection and the wave transmission coefficients data.

The results of the contemporary predictions of K_r and K_t are summarized within the tables 5.8, 5.9 and 5.10. The three tables differ each other for the training database: Tab. 5.8 aims to describe the ANN performance when trained on the common database to K_r and K_t (2'303 tests), while Tab.s 5.9 and 5.10 respectively synthesize the performance on the database of K_r (7'413) and of K_t (3'366).

In the first Tab. 5.8, the analysis is focused on the comparison between the effects of a single or a contemporary prediction of a same output (alternatively, K_r and K_t), by keeping constant the reduced training database.

Table 0.8 – Synthesis of the performance of the ANN trained on the common database to K_r and K_t (2'303). The results are separated for output parameter (K_r or K_t) and for "kind of prediction" (contemporary or single). A "contemporary" prediction indicates that the corresponding results are derived by training the ANN with the two outputs, while a single" prediction by training the ANN with only one output.

Prediction of K_r and K_t (common database #2'303)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.019 ± 0.002	0.990 ± 0.002	0.961 ± 0.008	0.08%
K_r	single	0.014 ± 0.001	0.994 ± 0.001	0.977 ± 0.004	0.1%
K_t	contemporary	0.023 ± 0.001	0.9971 ± 0.0006	0.988 ± 0.002	0.4%
K_t	single	0.023 ± 0.002	0.9973 ± 0.0004	0.989 ± 0.002	0.8%

The numerical indexes of Tab. 5.8 show an apparently good behavior of the ANN when applied to a contemporary prediction, as the plots of Fig. 5.23 confirm. The scatter is far limited and the degree of symmetry very high. However, observing these results, two negative considerations have to be pointed out:

- From a numerical point of view (see Tab. 5.8), the effect of a contemporary prediction for K_t does not generate any improvement of the performance, with respect to a single prediction. For K_r the effect is even detrimental, since the *rmse* sensibly increases from 0.014 to 0.019 and both *WI* and *R2* contextually decrease, when passing from a single to a contemporary prediction.

- The diagrams of Fig. 5.23, if compared to the corresponding ones of Fig. 5.1 left and Fig. 5.6 left, denote the effect of the extremely reduced number of tests and ranges of validity of the ANN when trained on the common database. As expected, having discarded all the tests referring to smooth impermeable slopes and seawalls (see Fig. 5.22), the K_r values are upper-limited to 0.65, and the most part of values is even included in the range 0.1÷ 0.4.

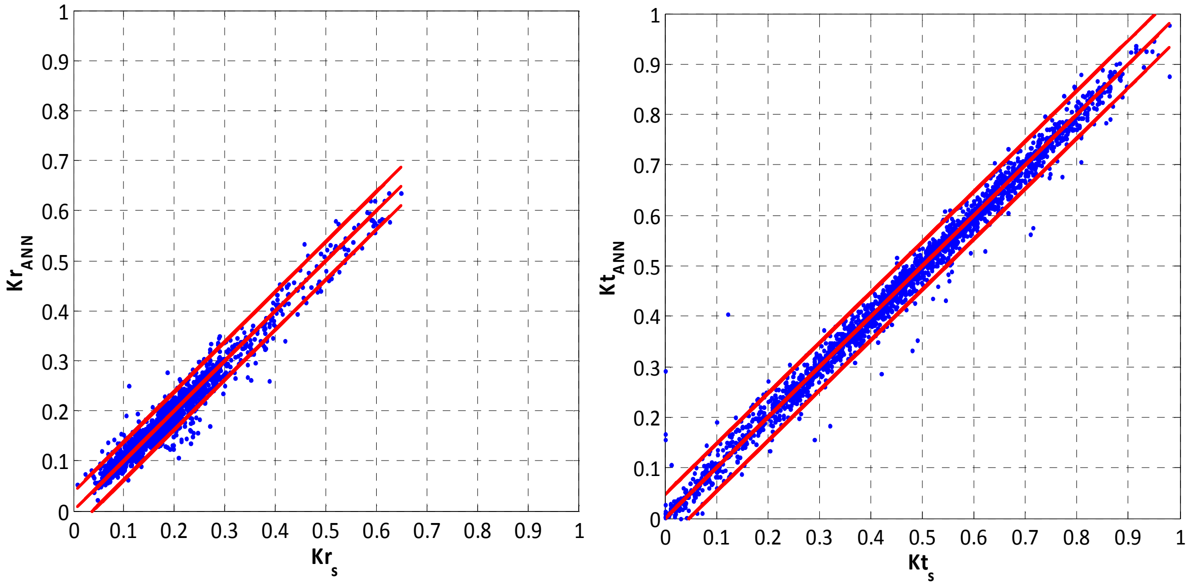


Figure 0.23 – Left: comparison among K_r predicted values ($K_{r,ANN}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); right: comparison among K_t predicted values ($K_{t,ANN}$ ordinate) and corresponding K_t experimental values ($K_{t,s}$, abscissa). The two outputs are contemporarily predicted by the trained ANN on the common database to K_r and K_t (2'303 tests).

The overall unsatisfactory effects and the serious limitations associated to the development of a contemporary prediction on the common narrower database, has leaded to the necessity to carry out another kind of test, described hereinafter.

The following Tables 5.9 and 5.10 provide a comparison between the two outputs K_r and K_t, when the ANN is trained with the two outputs and on, respectively, the complete databases of K_r and K_t. This second kind of comparison aims to show the results of a contemporary prediction when the ANN is forced to predict two outputs knowing the complete target values of just one of the two outputs themselves. This test was carried out in order to understand if the knowledge of just one of the two targets is sufficient to "learn" the input-output patterns of both the targets.

Table 0.9 – Synthesis of the performance of the ANN trained on the database of K_r (7'413). The results are derived by training the ANN with the two outputs K_r and K_t, as indicated by the column "kind of prediction". The results are separated for output parameter (K_r or K_t).

Prediction of K_r (database of K_r #7'413)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.038 ± 0.003	0.990 ± 0.002	0.961 ± 0.008	2.0%
K_t	contemporary	0.035 ± 0.002	-	-	-

Table 0.10 – Synthesis of the performance of the ANN trained on the database of K_t (3'366). The results are derived by training the ANN with the two outputs K_r and K_t, as indicated by the column "kind of prediction". The results are separated for output parameter (K_r or K_t).

Prediction of K_t (database of K_t #3'366)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.0203 ± 0.0008	-	-	-
K_t	contemporary	0.028 ± 0.001	0.9964 ± 0.0003	0.986 ± 0.001	8.5%

The results of Tab.s 5.9 and 5.10 clearly demonstrate that it is impossible to expect a good result for two output parameters when providing the ANN with just one complete target and one partial. The impossibility to compute the *WI* and the R^2 values for the "incomplete" outputs (i.e. for K_t in Tab. 5.9 and for K_r in Tab. 5.10) reveal that the corresponding predictions are completely unreliable or even nonsense.

5.4.2 Contemporary predictions of K_r and q

In this section, the performance of the ANNs with the respect to the contemporary inclusion of both the output parameters K_r and q is investigated. The training database has been reduced to 2'065 tests at all (the 28% of the total amount of tests available for wave reflection, i.e. 7'413 tests, and the 17% of the total amount of tests available for wave transmission, i.e. 11'825 tests). The pie chart of Fig. 5.24 illustrates the reduced database and shows the distribution of the residual available tests: differently from Fig. 5.22, the distribution appears more homogeneous and assorted, a positive pre-requisite.

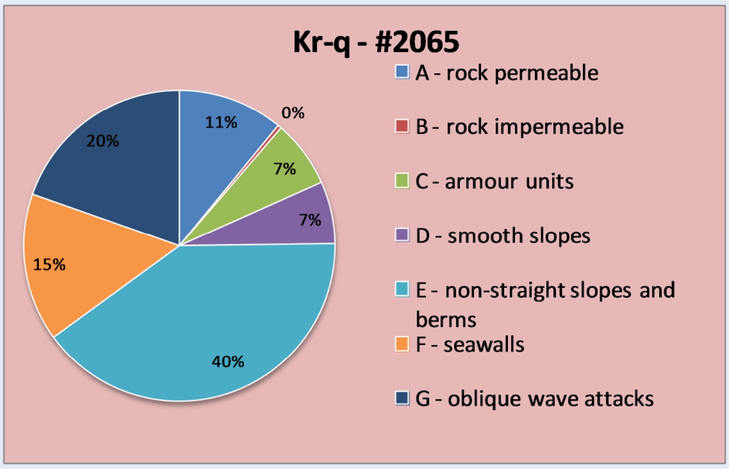


Figure 0.24 – Pie chart representing the distribution of the tests containing both the wave reflection coefficient and the wave overtopping discharge data.

The results are reported adopting the same scheme of the previous Paragraph 5.4.1. Table 5.11 describes the ANN performance when trained on the common database to K_r and q

(2'065 tests), while Tables 5.12 and 5.13 respectively synthesize the performance on the database of K_r (7'413) and of q (11'825).

In the first Tab. 5.11, the analysis is focused on the comparison between the effects of a single or a contemporary prediction of a same output (alternatively, K_r and q), by keeping constant the reduced training database. The following Tab.s 5.12 and 5.13 provide a comparison between the two outputs K_r and q, when the ANN is trained with the two outputs and on, respectively, the complete databases of K_r and q.

Similarly to the previous application, from Tab. 5.11 it is evident that the contemporary prediction of two outputs, even on the common database where both the targets are completely known, leads to a worsening of the performance, for both K_r and q.

Tab.s 5.12 and 5.13 remark again that the contemporary prediction of two outputs derived from an ANN trained on a database affected by an incomplete set of target, is impossible.

Table 0.11 – Synthesis of the performance of the ANN trained on the common database to K_r and q (2'065). The results are separated for output parameter (K_r or q) and for "kind of prediction" (contemporary or single). A "contemporary" prediction indicates that the corresponding results are derived by training the ANN with the two outputs, while a single" prediction by training the ANN with only one output.

Prediction of K_r and q (common database #2'065)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.034 ± 0.002	0.9939± 0.0009	0.976 ± 0.003	2.0%
K_r	single	0.029 ± 0.004	0.996± 0.002	0.983 ± 0.007	1.8%
$q>10^{-6}$	contemporary	0.045 ± 0.001	0.984 ± 0.001	0.940 ± 0.004	1.7%
$q>10^{-6}$	single	0.042 ± 0.004	0.987 ± 0.004	0.95 ± 0.01	1.7%

Table 0.12 – Synthesis of the performance of the ANN trained on the database of K_r (7'413). The results are derived by training the ANN with the two outputs K_r and q as indicated by the column "kind of prediction". The results are separated for output parameter (K_r or q).

Prediction of K_r (database of K_r #7'413)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.036 ± 0.002	0.991 ± 0.001	0.966 ± 0.004	3.1%
$q>10^{-6}$	contemporary	-	-	-	-

Table 0.13 – Synthesis of the performance of the ANN trained on the database of q (11'825). The results are derived by training the ANN with the two outputs K_r and q, as indicated by the column "kind of prediction". The results are separated for output parameter (K_r or q).

Prediction of q (database of q #11'825)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	contemporary	0.044 ± 0.005	-	-	-
$q>10^{-6}$	contemporary	0.053 ± 0.004	0.968 ± 0.006	0.89 ± 0.02	3.4%

The qualitative analysis of the results, provided by the diagrams of Fig. 5.25, enhances from one hand, the wider field of validity for K_r (with respect to the diagram of Fig. 5.23, left); from the other hand, the shortage of available tests which contemporarily contain the information on K_r and q. This latter issue is particular evident for q, if comparing the plot to the right of Fig. 5.25 with Fig. 5.9, left.

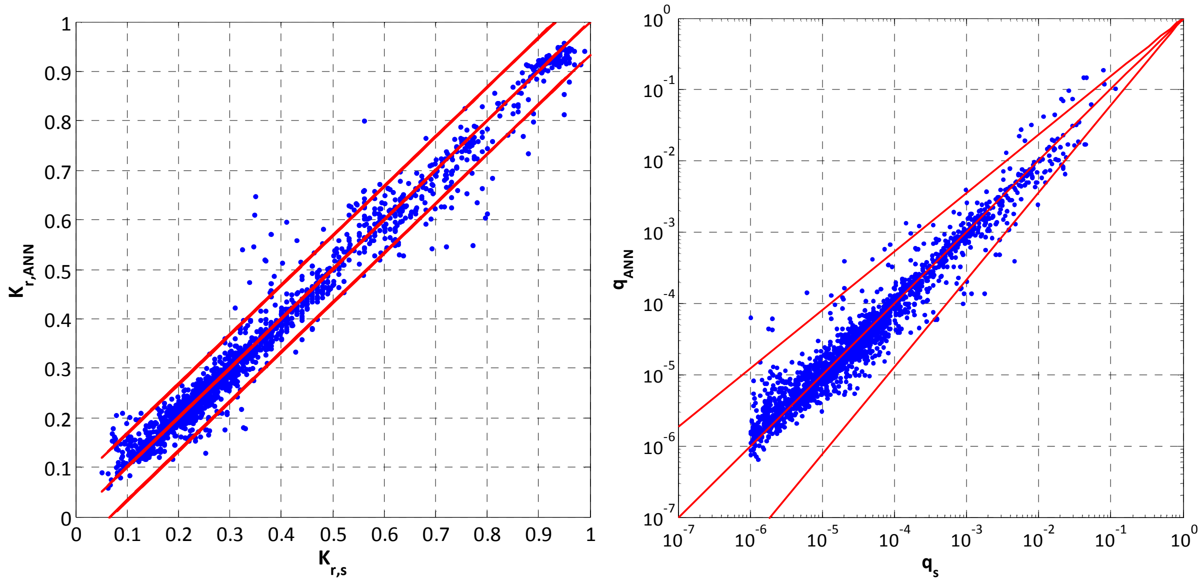


Figure 0.25 – Left: comparison among K_r predicted values ($K_{r,ANN}$ ordinate) and corresponding K_r experimental values ($K_{r,s}$, abscissa); right: comparison among q predicted values (q_{ANN} ordinate, logarithmic scale) and corresponding q experimental values (q_s, abscissa, logarithmic scale). The two outputs are contemporarily predicted by the trained ANN on the common database to K_r and q (2'065 tests).

5.4.3 Synthesis and conclusions on contemporary predictions

Within the sections 5.4.1 and 5.4.2, the performance of the ANN applied to the contemporary prediction of either K_r and K_t or K_r and q has been presented and discussed.

This last paragraph aims to draw some common conclusions about the matter of the contemporary predictions. For this purpose, Tab. 5.14 is displayed. This table collects the results obtained for each output parameter when training the ANN on the corresponding complete database, and varying the number of output neurons. The table is therefore organized into three sections, each of them dedicated to one of the three output parameters and the corresponding databases. Within each section, the results obtained by training the ANN with a single output neuron or two are synthesized. In case of a single prediction, the results are the “optimized” ones, displayed in Tab. 5.1.

The final purpose of Tab. 5.14 is to analyze the effects of a contemporary prediction on the original database in comparison to the results derived from the single prediction (please, note that the difference between Tab. 5.14 and Tab.s 5.8 and 5.11 is the database: in Tab. 5.14 the database is the original one, while in Tab.s. 5.8 and 5.11 the databases are the narrower ones common to K_r and K_t or K_r and q).

Table 0.14 – Synthesis of the performance of the ANN trained on, respectively, the database of K_r, K_t and q. Each section of the table presents the results associated to one output parameter and derived by training the ANN with the single output (single prediction), or with two outputs (contemporary prediction).

Prediction of K_r (database of K_r #7'413)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_r	single	0.038 ± 0.009	0.992 ± 0.008	0.97 ± 0.03	3.9%
K_r	contemporary with K_t	0.038 ± 0.003	0.990 ± 0.002	0.961 ± 0.008	2.0%
K_r	contemporary with q	0.036 ± 0.002	0.991 ± 0.001	0.966 ± 0.004	3.1%
Prediction of K_t (database of K_t #3'366)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
K_t	single	0.029 ± 0.009	0.996 ± 0.005	0.98 ± 0.02	8.4%
K_t	contemporary with K_r	0.028 ± 0.001	0.9964 ± 0.0003	0.986 ± 0.001	8.5%
Prediction of q (database of q #11'825)					
Output	Kind of prediction	*RMSE*	*WI*	R^2	# large errors (%)
$q \geq 10^{-6}$ m^3/(sm)	single	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%
$q \geq 10^{-6}$ m^3/(sm)	contemporary with K_r	0.053 ± 0.004	0.968 ± 0.006	0.89 ± 0.02	3.4%

The results of Tab. 5.14 allow concluding that, with the exception of the application on K_t, the adoption of a contemporary prediction damages the overall performance, instead than improving it. The worst condition is observed when training the ANN on the database of q and contemporarily predicting q and K_r. The ANN performs constantly better when working with only one output neuron.

In conclusion, at the present state, the contemporary predictions cannot be developed. The main cause is individuated in the shortage of tests reporting the datum of more than one output parameter. Another point could be the necessity to modify the ANN architecture and include more hidden layers with less neurons.

However, the discouraging outcomes obtained also on the narrower databases of common tests, where the single predictions resulted always better (see Tab. 5.8 and 5.11), could induce to think that the solution of a single output neuron is to be preferred.

5.5 Preliminary investigation about the application of a "logic" classifier for the pre-screening of the wave overtopping tests

The analysis of the prediction of the wave overtopping discharge (see Paragraph 5.2.3) has revealed that the ANN tends to overestimate the low values of q. It is supposed that the main cause of the bias is the definition of an arbitrary threshold (i.e. $q = 10^{-6}$ m^3/(sm)) to *a priori* discard part of the experimental database.

Verhaeghe et al. (2008) proposed the development a two-phase classifier-quantifier ANN in order to solve the matter of classify the tests to be delivered to the ANN, avoiding the arbitrary elimination of the data (for further details about Verhaeghe et al., 2008 work, see Paragraph 3.3.1). This solution would have also provided the ANN with an extended range of validity, so that the model should have been able to deal with "small" values of q, or, at least, to be able to classify them as "insignificant".

The objective of this Paragraph is to directly apply such approach to the new ANN and verify its outcomes. This methodology could be referred as a “logic” classifier-quantifier, because of the Boolean nature of the outputs of the classifier. The application of a “logic” classifier could be considered as the starting point for the optimization process of the two-phase network developed in this work and presented in the Chapter 6.

The methodology by Verhaeghe et al. (2008) has been followed step-by-step. It will not be described in the present section, since it has been already described in the Paragraph 3.3.1.

This section is organized into two parts: the Paragraph 5.5.1 illustrates the development of the logic classifier and the outcomes of the first phase of the classifier-quantifier approach. The Paragraph 5.5.2 presents and discusses the quantitative results of the final model, i.e. of the second phase quantifier ANN working in series at the classifier ANN.

5.5.1 Optimization of the logic classifier

Similarly to Verhaeghe et al. (2008), an artificial extended database of identically zero values of q was created. This extended database should provide the classifier ANN with a more equally distribution of data, being otherwise affected by a shortage of small or zero data (see Fig. 5.8).

The classifier ANN was therefore trained on the extended database: its architecture is completely similar to the one of the optimized standard ANN (see Paragraph 4.4), with the exception of the output neuron. Indeed, the classifier is forced to produce a logic output, instead of a quantitative estimation of q: this mean that the output values may be equal to either 0 (in case of a predicted negligible discharge, i.e. $q_{ANN} < 10^{-6}$ m^3/(sm)) or 1 (in case of $q_{ANN} \geq 10^{-6}$ m^3/(sm)). For simplicity, in the following the negligible q values will be said to belong to the “class -1” and the non-negligible to the “class +1”.

The bootstrapping was applied also for the classifier. Based on the results of the sensitivity analysis carried out to define the sufficient/optimal number of re-samples (see Paragraph 4.4.1), 50 bootstrap resamplings have been performed. Following the indications by Verhaeghe, three criteria were tested for the definite classification of a test as in the class -1 or +1:

- criterion 50%: the 50% quantiles of the distribution of the 50 bootstrap resamplings are adopted; this means that, if the ANN has settled a test 50%+1 times in the class -1 (i.e., for 26 out of 50 resamplings), it is assumed as “negligible” and discarded. Otherwise, it is passed to the quantifier (second phase);
- criterion 80%: the 80% quantiles are adopted, i.e. a test is discarded if the ANN settled it 80%+1 times in the class -1 (41 out of 50 resamplings).
- criterion 90%: the 90% quantiles are adopted, i.e. a test is discarded if the ANN settled it 90%+1 times in the class -1 (46 out of 50 resamplings).

The criteria 80% and 90% were investigated for safety reasons, i.e. to reduce the risk of ANN underestimations, avoiding that a test that actually belongs to the class +1 is discarded.

Table 5.15 collects the outcomes of the three tested criteria for the logic classifier. Being the ANN outputs logical values, the performance is reported in terms of percentages of misclassified data, and in details:

- overall misclassifications (column “% wrong tot”), i.e. data which the ANN has wrongly attributed to either class -1 or class +1;

- underestimations (column “% wrong class -1”), i.e. data which the ANN has wrongly attributed to class -1;
- overestimations (column “% wrong class +1”), i.e. data which the ANN has wrongly attributed to class +1.

In order to stress the improvement (or worsening) carried out by the adoption of the criteria 80% and 90%, the variation of performance with respect to the criterion 50% are displayed in green colour.

Table 0.15 – Results of the logic classifier, according to the adopted criterion. The column “% wrong tot” shows the overall % misclassified tests (i.e. the tests wrongly attributed either to class -1 or class +1); the column “% wrong class -1” shows the misclassified tests wrongly attributed to class -1; the column “% wrong class +1” shows the misclassified tests wrongly attributed to class +1. For criteria 80% and 90%, the variation with respect to criterion 50% is displayed.

Logic classifier of *q* (#11’825)			
Criterion for the classification	% wrong tot $(q_{ANN} \neq q_s)$	% wrong class -1 $(q_{ANN} < q_s)$	% wrong class +1 $(q_{ANN} > q_s)$
50%	5.5%	3.17%	2.4%
80% *(Δ respect to crit. 50%)*	8.0% *(+44%)*	3.20% *(+0.9%)*	4.8 % *(+103%)*
90% *(Δ respect to crit. 50%)*	8.7 % *(+57%)*	2.7% *(-15%)*	6.0% *(+155%)*

From Table 5.15, it is pretty evident that only the reduction of underestimated tests, carried out by criteria 80% and 90%, is nearly negligible, against a sensible increase of the overall misclassifications (respectively, +44% and +57%). Actually, only criterion 90% reduces the underestimations of the 15%, while criterion 80% unexpectedly even increases them of 0.9%.

Considering the results of this sensitivity analysis, the criterion 80% is completely useless. Differently, criteria 50% and 90% could be tested as upstream filters for the second phase quantifier ANN, in place of the fixed arbitrary threshold. The employability of these classifiers is verified in the next Paragraph 5.5.2.

5.5.2 Outcomes of the logic classifier-quantifier

This paragraph shows the results of the optimized new ANN working in series at the logic classifier. This way, the new ANN is employed as a second-phase quantifier, which receives as inputs the tests, which, according to the first-phase logic classifier, have been attributed to class +1. As anticipated, in this analysis only the classifier 50% and 90% are considered.

The quantifier performance – working downstream the two different classifiers 50% and 80% – is reported in the usual terms of quantitative error indexes (Tab. 5.16) and graphical diagrams displaying the distribution of the ANN predictions against the experimental values (Fig. 5.26).

Table 0.16 – Synthesis of the performance of the ANN applied to the wave overtopping working in series after a logic-classifier. Two different criteria for the classifier are investigated. The results are compared to the single ANN applied without the classifier.

Prediction of *q* from the classifier-quantifier				
Type of Classifier	*RMSE*	*WI*	R^2	# large errors (%)
50%	0.048 ± 0.004	0.976 ± 0.005	0.91 ± 0.02	4.2%
90%	0.056 ± 0.005	0.965 ± 0.007	0.87 ± 0.02	6.7%
No classifier	0.045 ± 0.003	0.978 ± 0.004	0.92 ± 0.01	1.9%

Both the quantitative and qualitative analysis show a serious worsening of the accuracy, when applying the first-phases classifiers. The main effect induced by the classifier 50% is an increase of the number of large errors, which duplicates (see Tab. 5.16) and which is reflected by a larger scatter in the distribution of the predictions (Fig. 5.26, left). However, the threshold value q=10^{-6} m^3/(sm) is pretty well reproduced, as it can be appreciated by Fig. 5.26 left, which displays no experimental q_s<10^{-6} m^3/(sm).

On the contrary, the classifier 90% generates an enormous scatter (Fig. 5.26, right) and demonstrates not to be able to catch the threshold, since many q_s<10^{-6} m^3/(sm) are detected. The inclusion itself of some (but not all) values q_s<10^{-6} m^3/(sm) in the database would not represent a problem, if the ANN proved to be able to deal with them and reduced the overestimation error. Instead, in Fig. 5.26 right the bias is still present and the scatter sensibly increased, reflecting the effects of the tripled number of large errors (Tab. 5.16) and the worsening of all the error indexes.

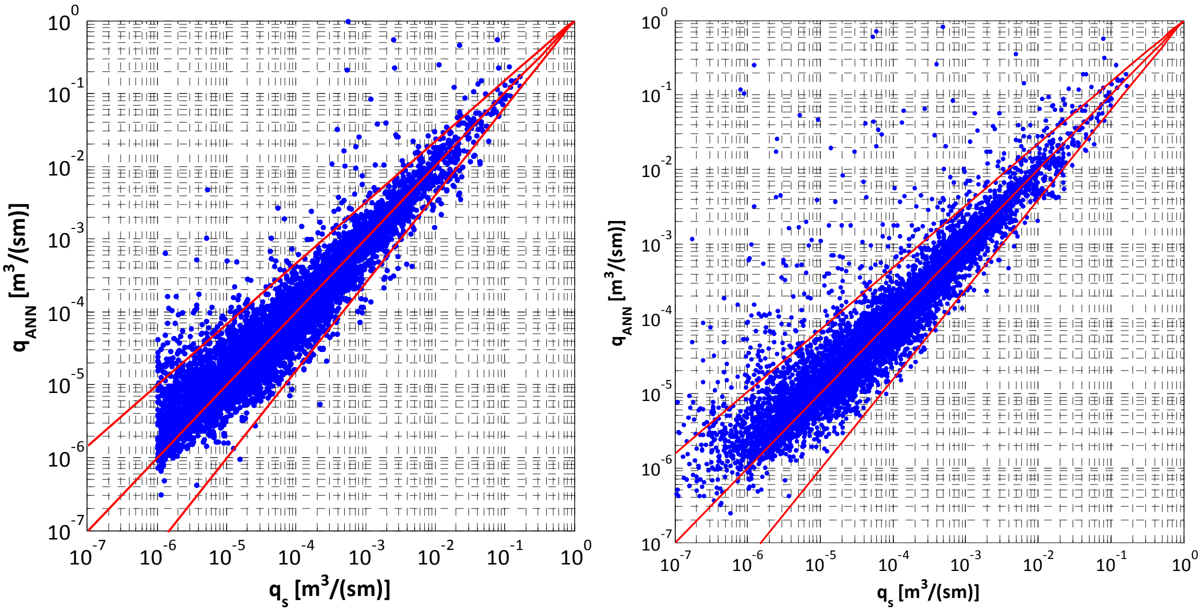


Figure 0.26 - Comparison among q predicted values (q_{ANN} ordinate, logarithmic scale) and corresponding q experimental values (q_s, abscissa, logarithmic scale). Left panel: ANN predictions derived downstream from the 50% classifier; right panel: ANN predictions derived downstream from the 90% classifier.

Then, the 50% classifier does not produce any significant improvement, with the exception of the actual capability to distinguish between “significant” and “neglecting” discharge which allows an extension of the field of validity of the ANN. Nevertheless, the increased scatter and the unresolved matter of the biased predictions do not allow adopting this methodology.

These problems are even more important when considering the 90% classifier, whose implementation sharply affects the ANN performance.

Finally, in this work, the logic classifier will not be adopted, and a different methodology *ad-hoc* will be developed (see Chapter 6).

5.6 Discussion and conclusions about the results of the new ANN

This chapter presented the results of the advanced ANN able to predict the overtopping discharge q accurately, the wave transmission coefficient K_t and the wave reflection coefficient K_r for a wide range of (complicated) structure geometries and wave attacks.

The results of the ANN, with reference to the parameters q, K_r and K_t, were analyzed and the comparison among predicted and experimental values was carried out, revealing that the predictions are satisfactorily accurate, providing values of the mean square error in the range of [0.03; 0.05]. The analysis of the distribution of the errors underlined the importance of the homogeneity and extension of the database. The accuracy of the ANN predictions for each of the three parameters was compared to existing formulae available in literature and existing ANNs already developed for each of the single process.

A specific study about the possibility to contemporary predict all the output parameters was carried out. The unsatisfactory results, principally attributed to the shortage of available tests reporting three, or at least two, output data, leaded to the impossibility to follow this path.

So far, the prediction of q is limited to the values larger than the fixed threshold 10^{-6} $m^3/(sm)$, according to the work and the considerations pointed out by Van Gent et al. (2007). An attempt to develop a logic classifier to replace the arbitrary threshold and enlarge the field of validity of the overtopping ANN was investigated but discarded because of an observed worsening of the overall performance. However, the two-phase approach combined to the enlargement of the overtopping database with tests of modest or null values of q could improve the performance of the ANN. For this purpose, the effects of introducing a different type of classifier-quantifier will be verified in the final Chapter 6 of this work.

6. OPTIMIZED PREDICTION OF WAVE OVERTOPPING DISCHARGE: A TWO-PHASE CLASSIFIER-QUANTIFIERS ANN

6.1 Introduction and aim

The optimized ANN for the prediction of the wave overtopping was trained against a selected database of 7'716 tests including only the data reporting an observed overtopping discharge q>10^{-6} m^3/(s·m). This database was adopted to carry out the sensitivity analysis finalized to the definition of the input set (see Paragraph 4.3) and the most suitable architecture (see Paragraph 4.4).

Following Van Gent et al. (2007), the "small values" of q have been excluded from the calibration and validation processes of the ANN, since "*…the proportion of erroneous data with q=0 was clearly higher* (in CLASH database) *than data with q>0. This is to some extent caused by different definitions of q=0 in different test programmes. For instance, it was found that q=0 can actually be a test with q≤1 l/s/m in large-scale tests while in small-scale tests q=0 can be a test with q≤0.001 l/s/m.*"

However, this approach imposes an "arbitrary" threshold value (i.e. 10^{-6} m^3/(sm)) to distinguish between "significant" and "non-significant" overtopping and reduces the field of validity of the ANN itself. Indeed, it is not clear which kind of outcome the ANN would provide if subdued to out-of-range input parameters, while a response of "non-overtopping" would be interesting for design purpose.

The attention to the "small" values of q is justified by the matter of defining the "tolerable" overtopping discharge. Indeed, "*most sea defense structures are constructed primarily to limit overtopping volumes that might cause flooding*", however "*designers and owners of these structures must also deal with potential hazards form overtopping. This requires that the level of hazard an its probability of occurrence be assessed, allowing appropriate action plans to be devised to ameliorate risks arising from overtopping*" (EurOtop Manual, 2007).

The Verhaeghe (2008) approach proposed the use of a classifier-quantifier ANN and the extension of the database to include additional zero values (see xx). The work was based on the incorrect assumption that all the values set to zero in CLASH database are really "zeros", and the extension of the database is likely to be affected by the unreliability associated to the original tests.

Our aim is to offer an ANN model which is able to represent also the cases of "small" overtopping, avoiding to affect the overall performance and reliability connected to the prediction of "large" overtopping cases.

The here purposed methodology consists in a two-phase "classifier-quantifiers" model, which adopts the already optimized ANN architecture for both the first-phase classifier and for the second-phase quantifiers. According to the classifier output, the data are passed to two distinct second-phase quantifiers, which respectively process the "small" and "large" values of q.

The first-phase classifier ANN is actually a quantifier that produces a numerical evaluation of q, instead of a logic value. This ANN is trained also on the zero values of the database, “artificially” modified (but not extended) to be very small values (down to 10^{-12}).

In order to assess the reliability of the observed zero overtopping tests – and therefore solve the matter of non-homogeneity of the zero values pointed out by Van Gent et al., 2007 – each of the zero tests modified and employed in the classifier training process is checked through a double comparison with a recent literature formula and the ANN predictions themselves.

We are going to show that the combination of a modified database and the adoption of two differently trained quantifier ANNs provides the user with a robust and efficient advanced prediction.

A first section (Paragraph 6.2) of this Chapter is dedicated to the analysis of the ANN performance against the prediction of the “small” (Paragraph 6.2.1) and the zero (Paragraph 6.2.2) values of q, in order to evaluate the ANN response out of the training ranges.

Then, Paragraph 6.3 presents the assessment of the reliability of the zero values and discusses the several solutions investigated to “modify” the zeros in order to be included in the training database.

Finally, Paragraph 6.4 illustrates the definitive “classifier-quantifiers” model: the definition of the classifier ANN of the first-phase is shown in Paragraph 6.4.1, while the two quantifier ANNs of the second-phase are presented in Paragraph 6.4.2. The overall performance of this new approach is discussed in Paragraph 6.4.3.

6.2 Prediction of “small” and zero values of overtopping

Aim of this section is to verify the ANN capability to deal with “small” or zero values of q. The ANN is tested against a wider part of the wave overtopping database, including also the values of measured $q \leq 10^{-6}$ m^3/(sm) and the cases of zero overtopping.

It is important to introduce that the inclusion of the zero values of the CLASH database, whatever the ANN, would lead to a totally biased and unreliable prediction. Furthermore, being the wave overtopping prediction based on the logarithmic transformation of the targets values of q, in case of inclusion of zeros, the logarithm of 0 would equal $-\infty$. To overcome this problem, the logarithmic transformation should be eliminated, leading to a further worsening of the performance. In order to stress this issue, Figure 6.1 reports the results of such ANN, trained also on the zeros. As expected, the plot shows a meaningless response of the model. The error indexes and the confidence intervals were not computed because of the hollow results.

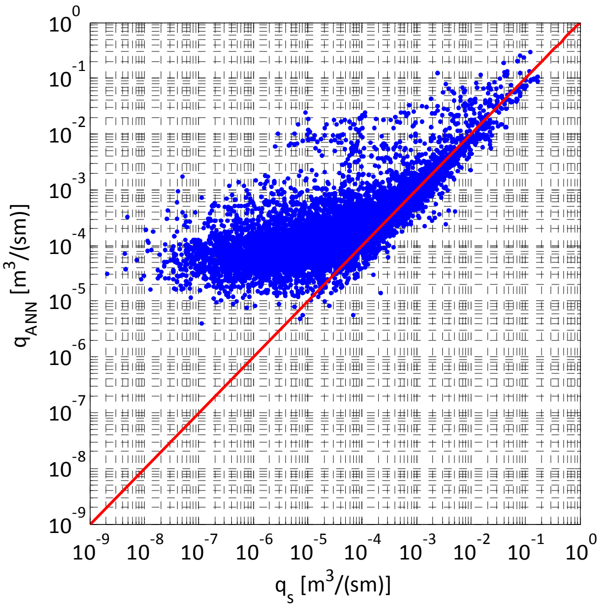


Figure 0.1 - ANNs predictions of the overtopping discharge (q_{ANN}, ordinate) vs the corresponding experimental values (q_s, abscissa). The bisector represents the perfect correspondence q_{ANN}=q_s, while the external bands the 95% confidence intervals. ANN on all the values of q, including the zero-overtopping tests.

The cause of this shortcoming is the ANN incapability to reproduce the input-output pattern when, despite the input conditions vary, the targets all exactly correspond to the same value, i.e. zero. In other terms, like each numerical tool or equation based on a data-fitting process, an ANN predicts values with continuity, and there is no way to fit a "clustering" of data on a same value. This phenomenon could be analytically represented by a horizontal line, where the abscissa varies to represent the different input conditions, while the ordinate is kept constant, being identically zero.

For this reason, the zeros cannot be included in the training database, except in case of substitution of the zeros themselves with "very low" values (see Paragraph 6.3). At the contrary, all the "small" non-zero values of q (i.e. $0 < q < 10^{-6}$ m^3/(sm)) could be introduced in the training set. Paragraph 6.2.1 is dedicated to the comparison of the performance between the "basic" ANN trained on $q > 10^{-6}$ m^3/(sm) and a "new" ANN trained on an enlarged database including all $q > 0$ m^3/(sm). In particular, this section is focused on the analysis of the effects of the introduction of "small" q values, in order to understand if the overall performance would be somehow affected.

In Paragraph 6.2.2 the performance of these two differently trained ANNs against the prediction of the zeros is discussed. The criterion to assess the ANNs performance relative to this application is the adoption of the threshold value $q = 10^{-6}$ m^3/(sm) and evaluating the ANNs responses in terms of "non-significant" (i.e. $q \leq 10^{-6}$ m^3/(sm)) or "significant" ($q > 10^{-6}$ m^3/(sm)).

6.2.1 Prediction of "small" values

The introduction of all the non-zero values (i.e. q > 0) determines an increase of the number of the available tests from 7'716 up to 8'783 tests (i.e. 1'067 tests, more than the 10% of the total amount of non-zero values refers to a "small value" of q).

The "optimized" ANN presented in Chapter 4 was trained on this extended database, following the methodology of the bootstrap resampling of the database (50 resamples). The quantitative results of this ANN (hereafter, ANN (b)) are shown in Table 6.1, in comparison with the performance related to the ANN trained on the "original" database (only tests with $q > 10^{-6}$ m^3/(sm), hereafter ANN (a)).

Table 0.1 – Performance of the ANN applied to the overtopping database comprehending only *q* > 10^{-6} values and the database extended to all non-zero values of *q*.

Prediction of *q* – 50 boots					
ANN (#training data)	*WF*	*RMSE*	*WI*	R^2	# Large errors (%)
a - q>10^{-6} (# 7'716)	yes	0.048± 0.005	0.974± 0.008	0.91± 0.03	198 (2.5%)
b - q>0 (# 8'783)	yes	0.050± 0.005	0.974± 0.009	0.91± 0.03	347 (3.9%)
a - q>10^{-6} (# 7'716)	no	0.045± 0.003	0.978± 0.004	0.92± 0.01	148 (1.9%)
b - q>0 (# 8'783)	no	0.048± 0.005	0.977± 0.007	0.92± 0.02	281 (3.2%)

From Tab. 6.1, we observe that the error indexes and the related standard deviations associated to ANN (b) are substantially equal to those of ANN (a), while there is a slight increase of the number of "large errors" (from 2.5% to 3.9%), which is due to the greater scatter associated to the predictions of the "small values".

More in details, Table 6.1 reports the performance of ANN (a) and ANN (b) obtained by either including or excluding the Weight Factors (*WF*) within the bootstrap resampling of the database. It is important to remark again that the "exclusion" of the *WF* regards the process of bootstrapping: indeed, the data reporting *RF* or *CF* = 4 have been anyway discarded from the training process (see Paragraph 4.4.4).

Despite the results of the sensitivity analysis which demonstrated that the inclusion of the *WF* generally generates a worse performance (see Paragraph 4.4.4 and Tab. 4.9), this approach has been tested again in order to follow the recommended methodology (Van Gent et al., 2007; Verhaeghe et al., 2008). Indeed, these authors suggest including the *WF* in order to take into account the "more reliable" and "less complex" structures in the training process.

However, as it can be appreciated from Table 6.1, for both ANN (a) and ANN (b) the exclusion of the *WF* carries out a slight improvement of performance also within this application, both regarding the values of the error indexes and the associated standard deviations. This improvement – which is more consistent for ANN (a) – involves also the number of "large errors".

This issue could be explained by taking into account two aspects: first, the definition of *RF* and *CF* is partially subjective and dependent on the evaluations carried out in different laboratories and facilities and different teams, and maybe it is not always consistent; second, the exclusion or the low-weighting of the most complex cases from the training – cases which instead would need to be "seen" by the ANN to be properly reproduced – affects the range of validity of the ANN and induces a reduced capability of generalization. At the contrary, by similarly weighting all the tests (or, better, not weighting them at all) should allow the ANN to be able to deal with a wider typology of cases.

Figure 6.2 qualitatively compares the performance of the two ANNs (respectively, ANN (a) to the left, ANN (b) to the right). From these plots, an increased scatter associated to ANN (b) is observed, which reflects the quantitative results reported in Tab. 6.1. Nevertheless, the inclusion of the "small values" clearly determinates a reduction of the overestimation of *q* in the range 10^{-6}÷10^{-5} m^3/(sm), where the predictions of ANN (a) tend to be biased, while the distribution of ANN (b) is sensibly more symmetric. Please note that Fig.6.2 left is the same as Fig. 5.9 left.

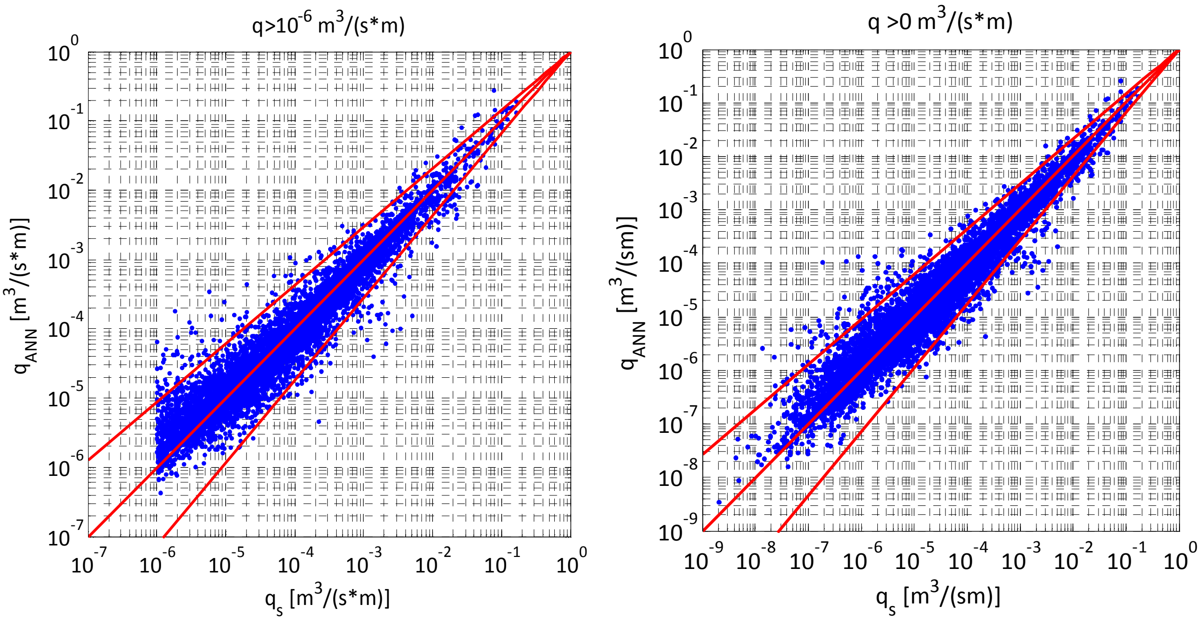


Figure 0.2 – ANNs predictions of the overtopping discharge (ordinate) vs the corresponding experimental values (abscissa). ANN trained only on experimental values of *q*>10^{-6} (to the left) and ANN trained on all the non-zero values of *q* (to the right).

Being the performance of ANN (b) generally comparable with the one associated to ANN (a), and being promising the reduction of the bias achieved with ANN (b), it is important to evaluate the behavior of these ANNs against the prediction of the zero values. If ANN (b) significantly reduced the overestimation of zeros, it could be preferred to ANN (a).

6.2.2 Prediction of zero values

As stated, the zero-overtopping cases cannot be included in the training process. However, they can be employed to test the ANNs in prediction. Adopting the CLASH-defined threshold value of *q* (Van Gent et al., 2007), an ANN prediction of a zero could be accepted as "correct" if it is lower than 10^{-6} and otherwise considered "wrong". This kind of output indeed would allow evaluating the ANN capability to recognize, at least, if the overtopping discharge is "significant" or "non-significant".

Then, both ANN (a) and ANN (b) (for the definition, see Paragraph 6.2.1) have been applied to the prediction of the zeros present in the CLASH database. These tests amount to 1'113 and represent nearly the 15% if compared to the database restricted to q > 10^{-6} (7'716) and the 12% to the database enlarged to q > 0 (8'783).

The results of the two ANNs are reported in Table 6.2, in terms of percentage of "wrong" predictions. Besides 10^{-6}, another threshold value (q = 10^{-4}) has been defined, in order to detect how many "wrong" predictions are greater than 10^{-6} of more than two orders of magnitude.

In Tab. 6.2 both the cases of inclusion and exclusion of *WF* are reported. Differently from Tab. 6.1, in this case the elimination of the *WF* induces a worsening of the performance, both for ANN (a) and ANN (b).

Focusing at the moment on the case of inclusion of *WF*, as expected, the percentage of "wrong predictions" is lower when applying ANN (b) instead of ANN (a), but its value (41%) is pretty high. Moreover, both the ANNs have predicted a non-negligible amount of values $q > 10^{-4}$, revealing a sensible overestimation of the experimental zeros.

On the other hand, it has to be noted that, being the absolute error defined as the difference among experimental (q_s) and predicted (q_{ANN}) values of q, and being the experimental values identically equal to zeros (q_s=0), the error

$$e = q_s - q_{ANN} = 0 - q_{ANN} = - q_{ANN}.$$

Eq. 0.1

In other terms, the error e results of the same order of magnitude of the prediction q_{ANN} itself, i.e. 10^{-4} m^3/(sm), two orders of magnitude lower than the average *rmse* characterizing the overall ANNs performance (~10^{-2}, see Tab. 6.1), which is in the natural scale of the values of q (see the definition of *rmse*, Eq. 4.3).

Table 0.2 – Prediction of 0-overtopping values, comparison among the ANN trained on the database comprehending only $q > 10^{-6}$ values and the ANN trained on the database extended to all $q > 0$.

Prediction of zero-overtopping (#1'113)			
ANN (training database)	*WF*	# Wrong >10^{-6} (%)	# Wrong >10^{-4} (%)
a (q>10^{-6})	yes	616 (55%)	71 (6%)
b (q>0)	yes	451 (41%)	57 (5%)
a (q>10^{-6})	no	626 (56%)	81 (7%)
b (q>0)	no	489 (44%)	71 (6%)

In order to show the datasets within the wrong predictions are most concentrated, Table 6.3 is reported. This table, besides the labels of the principal datasets, provides:

- the percentage of the measured zeros present in each dataset;
- the percentage of wrong predictions performed by ANN (b) for each dataset;
- the minimum value of observed $q > 0$ for each dataset.

From Tab. 6.3 it is evident that the zero values in the CLASH database – whatever is the dataset – were determined by different methodologies of the measurement or resolution or decision of the responsible to fix a "threshold", as further proof of the non-homogeneity of the zeros (see Paragraph 6.1 and Van Gent et al., 2007). For example, within the CLASH groups 103 and 110 the lowest values of observed q is of the order of magnitude of ~10^{-4}÷10^{-3}, while group 027 reports values of q down to ~10^{-8}.

This issue was at the basis of the choice to prefer the simple quantifier ANN to the two-phase classifier-quantifier ANN proposed by Verhaeghe et al., 2008 (see Paragraph 3.3.1). Within the work of Van Gent et al. (2007) it was stated that: "*an alternative approach was also considered but not adopted, namely a first NN (i.e. ANN) to predict whether for a specific situation there is overtopping or not, and a second NN to predict the actual overtopping discharge. [...]This is to some extent caused by different definitions of q=0 in different test programs*".

Table 0.3 – Report of the principal experimental datasets within the highest (>10^{-4}) or the most recurrent errors are concentrated.

Dataset label	CLASH group	% of 0 within the dataset	% of wrong 0-predictions	Min $q \neq 0$
A393-409	027	65%	18%	3.39E-8
A2275-2328	917	44%	58%	2.06E-7
B584-638	601	44%	62%	2.27E-7
B708-736	603;604	83%	97%	2.06E-6
C2140-2409	331	17%	87%	1.00E-6
D89-185	226	36%	89%	1.60E-6
D1155-1174	103	25%	100%	1.36E-3
D1342-1374	110	39%	77%	1.20E-4
E325-418	109	32%	43%	1.14E-6
E531-700	-	69%	91%	1.00E-6
E1857-1937	101	26%	95%	3.65E-4
F296-685	113	31%	43%	1.50E-5

In order to clarify how much the different definition of the zeros affects the interpretation of the ANN predictions, the formula by van der Meer et al. (2013) for the prediction of the wave overtopping (see Eq. 6.2) was applied to these experimental data.

$$q_{overtop}=\frac{0.026}{\sqrt{\tan\alpha}}\cdot\xi_{m-1,0}\cdot \exp\left(-\left(2.7\frac{0.226}{\xi^{3}_{m-1,0}\cdot H_{m0}\cdot\gamma_b\cdot\gamma_f\cdot\gamma_v\cdot\gamma_\beta}\right)^{1.3}\right)\cdot\sqrt{gH^3_{m0}}$$

with a maximum of

$$q_{overtop}=0.09\cdot \exp\left(-\left(1.5\frac{R_c}{\xi^{3}_{m-1,0}\cdot H_{m0}\cdot\gamma_f\cdot\gamma_\beta}\right)^{1.3}\right)\cdot\sqrt{gH^3_{m0}}\quad .$$

Eq. 0.2

The ANN predictions of the zeros were compared to the estimations of the same values provided by Eq. 6.2. From this comparison, it was established that:

- when both ANN and Eq. 6.2 overestimate a zero, providing a value of $q >> 10^{-6}$, the reliability of that test is supposed to be poor;
- in case of discordance between ANN and formula, a failing of one of the methods is suspected and the test is likely to be reliable.

By adopting this criterion, an amount of 148 tests (the 13% of the overall 1’113 zero-overtopping tests) was considered “unreliable”. Table 6.4 reports the number of experimental zero-overtopping tests present in each dataset and the corresponding number of tests that were found “unreliable”. As it can be appreciated from Tab. 6.4, the “unreliable” tests are mostly concentrated in the datasets C, D, E and F.

Table 6. Errore. Nel documento non esiste testo dello stile specificato.**.4 – Number and percentage of the zero-overtopping tests present in each dataset of the overall database and corresponding number of tests considered "unreliable".**

Dataset	# 0-tests in the dataset (%)	# 0-tests "unreliable" (%)
A	115 (13%)	3 (0.2%)
B	103 (8%)	4 (3.9%)
C	231 (18%)	23 (9.2%)
D	99 (7%)	39 (38%)
E	284 (14%)	38 (12%)
F	237 (11%)	41 (16%)
G	44 (5%)	0 (0%)

An example of this analysis is proposed in Figures 6.3 and 6.4, referring to the datasets F and D (respectively comprehending tests on seawalls and on impermeable slopes).

Figure 6.3 shows the predictions of both ANN (a) and ANN (b) in terms of the non-dimensional group $q/(g{\cdot}H^3_{m,0,t})^{0.5}$ as functions of the ratio $R_c/(H_{m,0,t}{\cdot}\gamma_f)$. In both the plots, the trend of Eq. 6.2 is represented through the continuous line.

Similarly, Figure 6.4 shows the ANN predicted values (ordinate) against the predictions for the same tests obtained with Eq. 6.2 (abscissa), being the continuous line the perfect correspondence. It is important to bear in mind that the actual experimental values of q are all equal to 0.

From both Figs. 6.3 and 6.4, it is evident that ANN (b) generates a slight, but not sufficient, improvement with respect to ANN (a). This improvement is more evident for dataset F (panel to the left of Fig. 6.4).

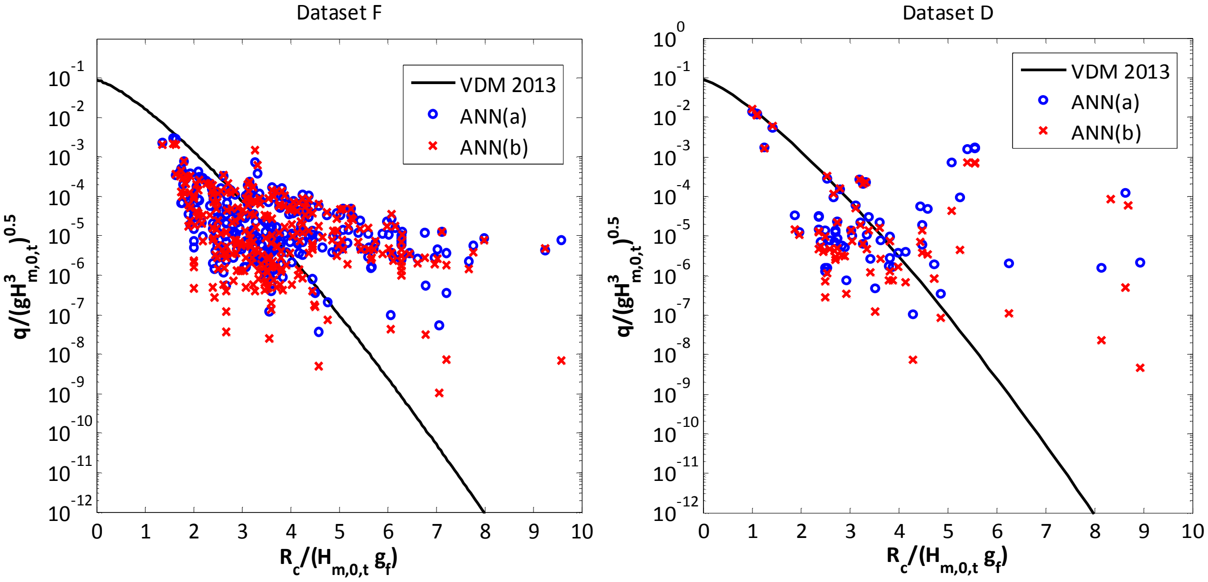


Figure 0.3 – Distribution of the ANNs predictions against van der Meer et al. (2013) formula (continuous lines); the predictions are presented as functions of the non-dimensional parameter Rc/(Hm,0,t·gf) (abscissa). ANN (a) is trained on the database comprehending only q>10^{-6} values; ANN (b) is trained on the database extended to all q>0. Panel to the left: 0-overtopping tests belonging to dataset "F" (seawalls); panel to the right: 0-overtopping tests belonging to dataset "D" (impermeable straight slopes).

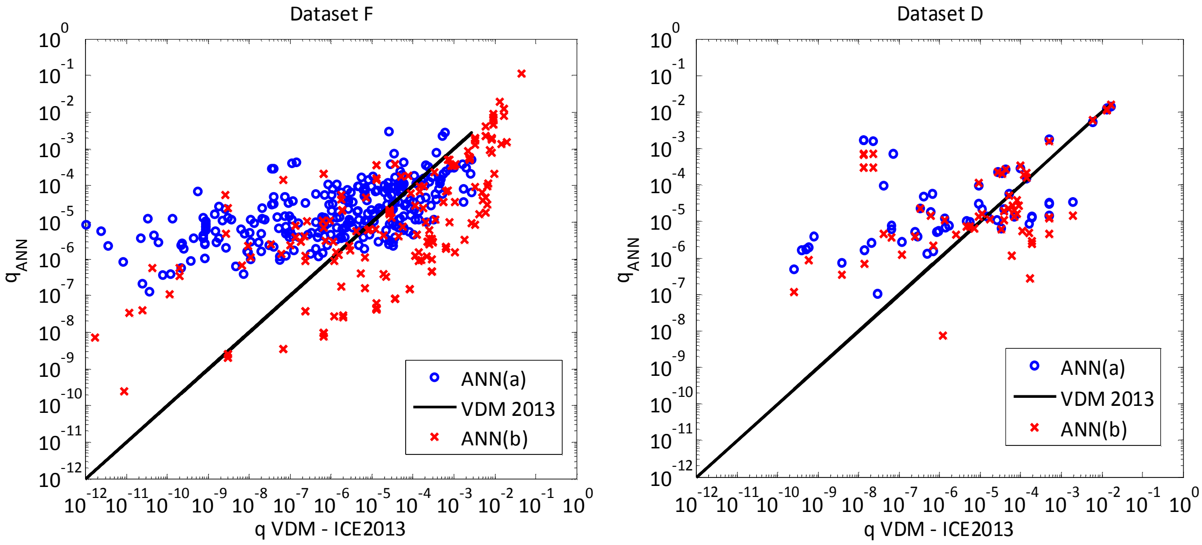


Figure 0.4 – ANN predictions of *q* (ordinate) as functions of van der Meer et al. (2013) predictions; ANN (a) is trained on the database comprehending only *q*>10^{-6} values; ANN (b) is trained on the database extended to all q>0. Panel to the left: 0-overtopping tests belonging to dataset "F" (seawalls); panel to the right: 0-overtopping tests belonging to dataset "D" (impermeable straight slopes).

Generally (see Fig. 6.3), for small values of $R_c/(H_{m,0,t}\cdot\gamma_f)$ (i.e. 1< $R_c/(H_{m,0,t}\cdot g_f)$ <4) there is a good correspondence among ANNs and formula, and both the methods provide estimations of *q* sensibly higher than 10^{-6}. Within this range, the formula nearly represents an upper envelope for the ANNs predictions, and therefore it could be concluded that the overestimations are more likely to be induced by an "inhomogeneous" definition of the zeros within the database.

Differently, for higher values of $R_c/(H_{m,0,t}\cdot g_f)$ (i.e. $R_c/(H_{m,0,t}\cdot g_f)$ >4), Eq. 6.2 effectively provides very low values of *q*, which are not consistent with the ANN predictions. For these cases, the ANNs seem to partially fail the prediction of the zeros, as it is confirmed by the plots of Fig. 6.4, where, especially for ANN (a), a large part of the ANN predictions stand above the line of the perfect correspondence with the formula.

In conclusion, this analysis has revealed that:

- based on the double comparison with ANN predictions and van der Meer et al. (2013) formula, part of the zeros of the database (148 data on 1'113) has to be considered "unreliable";
- from now on, the performance of the ANNs against the prediction of the zeros will be evaluated by excluding the 148 tests considered unreliable;
- the limits of the ANNs are only partially due to the non-homogeneous definition of the zeros, since both the ANNs (a) and (b) show an intrinsic tendency to overestimate the lowest values, with respect to van der Meer et al. (2013) formula;
- ANN (b), even if it is trained on a more extended database, including the "small values" of *q*, does not completely overcome the problem of representing the zeros.

The unsatisfactory improvement of ANN (b) in the prediction of the zeros, has leaded to the idea of further extend the training database, in order to include also the identically zero values. This idea crashes against the already discussed practical impossibility of training an ANN on targets equal to zero.

In the next Paragraph 6.3, the techniques developed to overcome this problem, mainly based on the "substitution" of the zeros with "very low" values, are described in details and discussed.

6.3 Replacing of the zero-overtopping values

The "replacing" of the zero-overtopping tests essentially consists in the substitution of the target values of *q* (equal to zero) with artificial values lower than the threshold q=10^{-6} m^3/(sm), i.e. with values "practically" recognized as "non-significant".

The replacement of the zero-overtopping tests involves the re-training of the ANN on a database which includes the modified tests reporting q = 0. In this case the total amount of data rises up to 9'896 (please, remember the tests presenting *RF* or *CF* = 4 are always excluded). Such a re-trained ANN will be referred hereinafter as ANN (c).

It is important to remark that the substitution of the zeros affects only the training phase. The original experimental zeros were of course kept, but none introduced in the training database.

The zero-values cannot be replaced with a fixed value (for example, q=10^{-9} m^3/(sm), which corresponds to the lowest registered value actually present in the database), but need to be proportionally re-scaled or re-distributed with continuity from 10^{-6} down to a variable lower limit. In fact, the ANN cannot deal with a fixed target value when the input set varies, whatever the target value is (see Paragraph 6.2).

Several methodologies of replacement were investigated, in order to optimize the ANN (c) performance. In the following, the three main attempts are reported:

i. Rescaling of the zeros, based on the ANN predictions: starting from the predictions of the zero-values performed by ANN (b) (named, hereafter, *x*), the following rescaling has been applied:

$$y=\frac{(d\text{-}c)\cdot x}{b\text{-}a}+\frac{bc\text{-}ad}{b\text{-}a},\quad \begin{cases} a=\min(x);\; b=\max(x) \\ c=10^{-12};\, d=10^{-7} \end{cases}$$

Eq. 0.3

where *y* is the rescaled target value which should replace the experimental target (i.e. 0). The "new" target values *y* can space between *d* and *c*, i.e. between 10^{-7} and 10^{-12}, while the values of x space between *a* and b (by definition of a and b).
Some modifications to the transformation applied with Eq. 6.3 were tested, by varying c and d: for example, c has been set to greater values (up to 10^{-9}) or to lower (down to 10^{-15}), or chosen as the maximum value between a fixed value (10^{-9}, 10^{-12}, 10^{-15}) and a itself.
Another variation consisted in the replacement of *x*, through rescaling, only for values of $x > 10^{-6}$, otherwise the $x < 10^{-6}$, have been kept.
Whatever the rescaling technique, this approach presumes to entrust the ANN (b) predictions entirely.

ii. Employment of the formula by van der Meer et al. (2013). In this case, Eq. 6.2 was directly applied to perform the "new" target values: *y*-values, therefore, correspond to the minimum between the outcomes of Eq. 6.2. No process of rescaling was implemented, but a simple substitution was applied. This approach presumes to entrust the van der Meer et al. (2013) predictions entirely.

iii. Combined application of the outcomes derived from ANN (b) and van der Meer et al. (2013) formula. The combination of the approaches should overcome the limits of each one applied individually. This methodology can be described by means of the following logical scheme (Fig. 6.5):

$$\begin{cases} I) & if\ x_{ANN(b)}>10^{-6}\ \ and\ \ x_{VDM}>10^{-6} \rightarrow\ y=NaN \\ II) & if\ x_{ANN\,(b)}\leq 10^{-6}\ and\ x_{VDM}>10^{-6}\rightarrow\ y=x_{ANN\,(b)} \\ III) & if\ x_{ANN(b)}>10^{-6}\ \ and\ \ x_{VDM}\leq 10^{-6}\ \rightarrow y=x_{VDM} \\ IV) & if\ x_{ANN(b)}\leq 10^{-6}\ \ and\ \ x_{VDM}\leq 10^{-6}\ \rightarrow y=max(x_{ANN(b)},x_{VDM}) \end{cases}$$

Figure 0.5 – Scheme representing the technique of "replacement" of the zero-values of the database by accounting of both ANN predictions and the estimations of *q* provided by van der Meer et al. (2013) formula.

The system of Fig. 6.5 indicates that when both ANN (b) and the formula provide values of $q > 10^{-6}$ (case *I*) the corresponding experimental zero-value is considered as unreliable and then eliminated from the training (following the same criterion described in Paragraph 6.2.2 for the assessment of the zero values reliability). When ANN (b) and the formula provide discordant results (i.e. one model predicts $q > 10^{-6}$ and the other one $q < 10^{-6}$, cases *II* and *III*), the minimum is preferred. Finally, when both the predicting methods provide $q < 10^{-6}$ (case *IV*) the maximum is preferred, to prevent an under-estimation bias of the ANN.

The schemes of Fig. 6.6 aim to represent the logical steps followed to shape and train the three different ANNs (c). As it is depicted, each ANN (c) is trained on a different modified "database" of 9'896 data obtained by the replacement of the zeros. For example, in case of ANN (c-i), the modified Database' was created based on the predictions obtained from ANN (b).

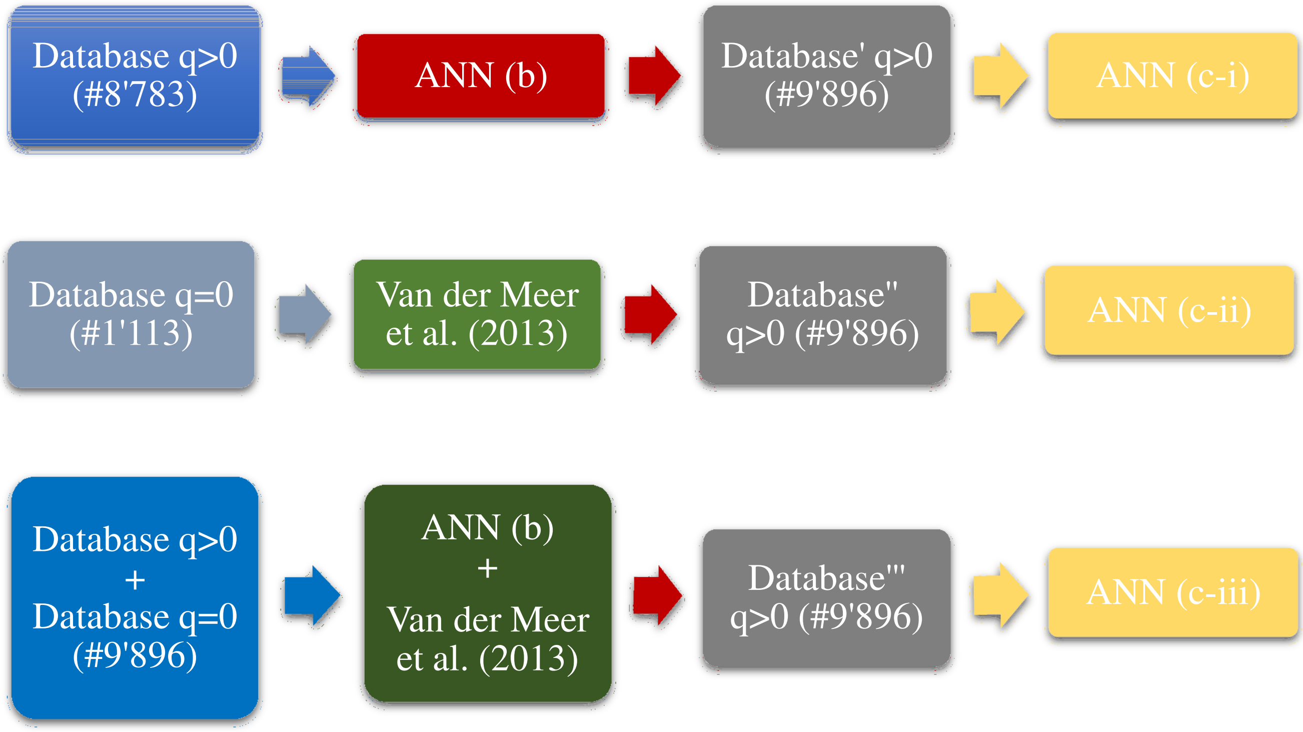


Figure 0.6 – Scheme representing the logical steps followed to create the "extended" database including the zero-overtopping values. Each scheme illustrates the adopted methodology to replace the zeros and the resulting ANN (c).

Independently from the technique of “replacement”, the substantial difference among ANN (c) and ANN (a) or ANN (b) is the training database. Once trained, each of the ANNs (c) can be applied to the prediction of whatever dataset, including the database of 8’783 non-zero tests (which corresponds to ANN (b) training database). In order to evaluate the overall performance of ANNs (c) and compare it to the previous application ANN (b), this aspect becomes paramount.

The following Paragraphs 6.3.1 and 6.3.2 are therefore dedicated to the analysis of performance of the three ANNs (c) regarding the prediction of both the zero (6.3.1) and non-zero values (6.3.2).

6.3.1 Prediction of the zero values

The results of the predictions of the zero values obtained with the three different ANN (c) are resumed in Table 6.5. These results can be directly compared to the ones presented in Table 6.2 and referring to ANN (a) and ANN (b), applications without *WF*. All the three cases of ANN (c) lead to an improvement in the ANN capability of predicting the zero values. In particular, ANN (c-ii) does not predict any value >10^{-4} and only a 12% of overestimations >10^{-6}. At the contrary, ANN (c-iii) seems to provide the lowest improvement, since it “only” halves the overestimations, if compared to ANN (b) (see Tab. 6.2).

Table 0.5 – Prediction of zero-overtopping values, comparison among the ANNs trained on all the available overtopping tests reporting RF and CF ≠ 4; the zero-overtopping values have been artificially replaced within the training phase according to different methodologies.

Prediction of zero-overtopping (#1’113)			
ANN	***WF***	**Wrong >10^{-6} (%)**	**Wrong >10^{-4} (%)**
c(i) (#9’896)	no	196 (18%)	7 (0.6%)
c(ii) (#9’896)	no	112 (12%)	0 (0.0%)
c(iii) (#9’896)	no	263 (23%)	39 (3.5%)

The performance of ANN (c-iii) was furtherly investigated by analyzing the predictions of the zeros relative to each dataset. Table 6.6 reports the percentage of “wrong” zero-predictions, similarly to Tab. 6.3, while Fig. 6.7 (similar to Fig. 6.2) shows the distribution of the predictions around the curve relative to van der Meer et al. (2013) for datasets D and F. The comparison with the previous application ANN (b) is provided in both Tab. 6.6 and Fig. 6.7, reporting the corresponding quantitative and qualitative results of Tab. 6.2 and Fig. 6.2, respectively.

Table 0.6 – Report of the principal experimental datasets within the highest (>10^{-4}) or the most recurrent errors are concentrated. Comparison among ANN (b), trained on all test reporting q>0 and ANN (c-iii), trained on all the available overtopping tests reporting RF and CF ≠ 4. For ANN (c-iii), the 0-overtopping values have been artificially replaced.

Dataset label	CLASH group	% of wrong zero-predictions	
		ANN (b)	ANN (c-iii)
A393-409	027	18%	9%
A2275-2328	917	58%	12%
B584-638	601	62%	12%
B708-736	603;604	97%	14%
C2140-2409	331	87%	15%
D89-185	226	89%	20%
D1155-1174	103	100%	100%
D1342-1374	110	77%	15%
E325-418	109	43%	33%
E531-700	-	91%	6%
E1857-1937	101	95%	15%
F296-685	113	43%	2%

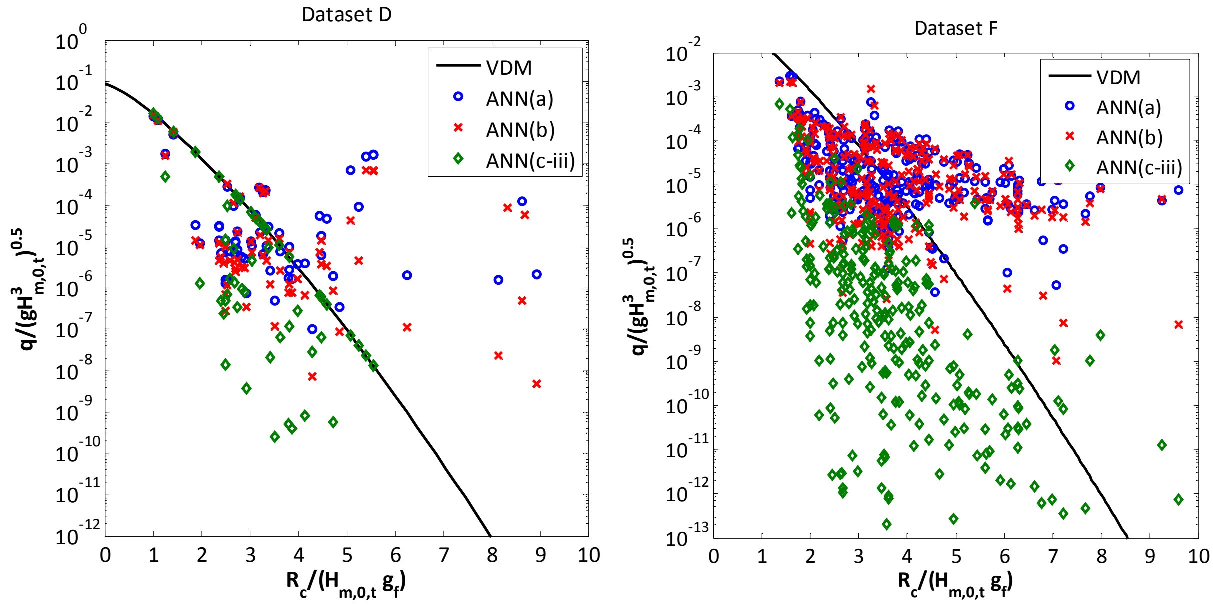


Figure 0.7 - Distribution of the ANNs predictions against van der Meer et al. (2013) formula (continuous lines); the predictions are presented as functions of the non-dimensional parameter *$R_c/(H_{m,0,t}{\cdot}g_f)$* (abscissa). Comparison among ANN (a) (trained on *q*>10^{-6} tests), ANN (b) (trained on *q*>0 tests) and ANN (c) (trained on all the available overtopping tests reporting RF and *CF* ≠ 4, being the 0 values replaced). Panel to the left: 0-overtopping tests belonging to dataset "F" (seawalls); panel to the right: 0-overtopping tests belonging to dataset "D" (impermeable straight slopes).

Both Tab. 6.6 and Fig. 6.7 show that, though ANN (c-iii) involves the least improvement in the predictions of the zeros, the progress in respect to ANN (b) is great. With the exception of dataset D1155-1174 (corresponding to group 103 of the original CLASH database), for which no advance is obtained, ANN (c-iii) not only reduces the number of overestimations but also the general scatter. The predictions are far less biased (see, in particular, plot to the right of Fig. 6.7) and, as expected, more "rationally" distributed along below the lines representing the curve of van der Meer et al. (2013). This latter aspect is a direct effect of the inclusion of the formula in the "replacement" of the zero-values.

6.3.2 Prediction of the non-zero values

In order to a establish if one of the ANNs (c) might represent a good compromise between the overall prediction of "significant" and "non-significant" overtopping, a quantitative comparison with ANNs (a) and (b) is required. Table 6.7 – to be compared to Tab. 6.1 – reports the performance of the three ANNs (c) applied to the overtopping database including only q>0 tests (i.e. predicting 8'783 values). For a more direct comparison, also the results relative to ANN (b) are reported. In this table, the number of "large errors" is missing for the ANNs (c) because considered unreliable. Indeed, a large error occurs nearly for each of the lowest values of *q*, being the error of the same order of magnitude of *q* itself.

From Tab. 6.7 it is clear that the reduction of the number of overestimations of zeros provided by ANNs (c) is obtained at the expense of a considerable worsening of the overall performance (i.e. of the prediction of all non-zero data). Moreover, the higher the reduction of zero overestimations (ANN (c-i) and ANN (c-ii)) the poorer the capability to represent the non-zero cases.

Table 0.7 – Indexes of performance relative to the prediction of the non-zero overtopping values (#8'783) performed by the three ANNs (c) trained on all the available overtopping tests reporting RF and CF ≠ 4 (# 9'896). Comparison with the performance of ANN (b) trained only on non-zero overtopping values (#8'783).

Prediction of *q*>0, (#8'783)					
ANN (# training database)	*WF*	*RMSE*	*WI*	R^2	# Large errors (%)
c(i) (# 9'896)	no	0.055±0.004	0.93±0.01	0.72±0.05	-
c(ii) (# 9'896)	no	0.051±0.009	0.95±0.1	0.82±0.9	-
c(iii) (# 9'896)	no	0.05±0.05	0.96±0.06	0.84±0.4	-
b (#8'783)	no	0.048± 0.005	0.977± 0.007	0.92± 0.02	284 (3.2%)

This issue is qualitatively confirmed by the plots of Fig. 6.8, which report the distribution of the predicted values (q_{ANN}) against the corresponding experimental values (q_s). From left to right and from top to bottom, the four plots of Fig. 6.7 respectively refer to the predictions obtained from ANN (c-i), ANN (c-ii), ANN (c-iii) and ANN (b). This last one – which is the same diagram of Fig. 6.2, panel to the left – has been repurposed here as a "benchmark" case.

Then, considering the "overall" performance, the best application has resulted ANN (c-iii), though the higher values of standard deviations and reduced indexes R^2 and *WI* (see Tab. 6.7) remark that none of the cases of ANN (c) can really predict with sufficient accuracy both the low values and give an overall good estimate of the non-zero values.

An ANN – such as each of ANNs (c) – affected by larger error within the prediction of significant overtopping has to be discarded. For design purpose and for security reasons, it is anyway preferable an overestimation bias relative to "non-significant" overtopping values, instead that an ANN providing a worse performance within the field of "significant" values.

So far, it can be concluded that it is not possible to adopt a single ANN to predict all the tests of the experimental database.

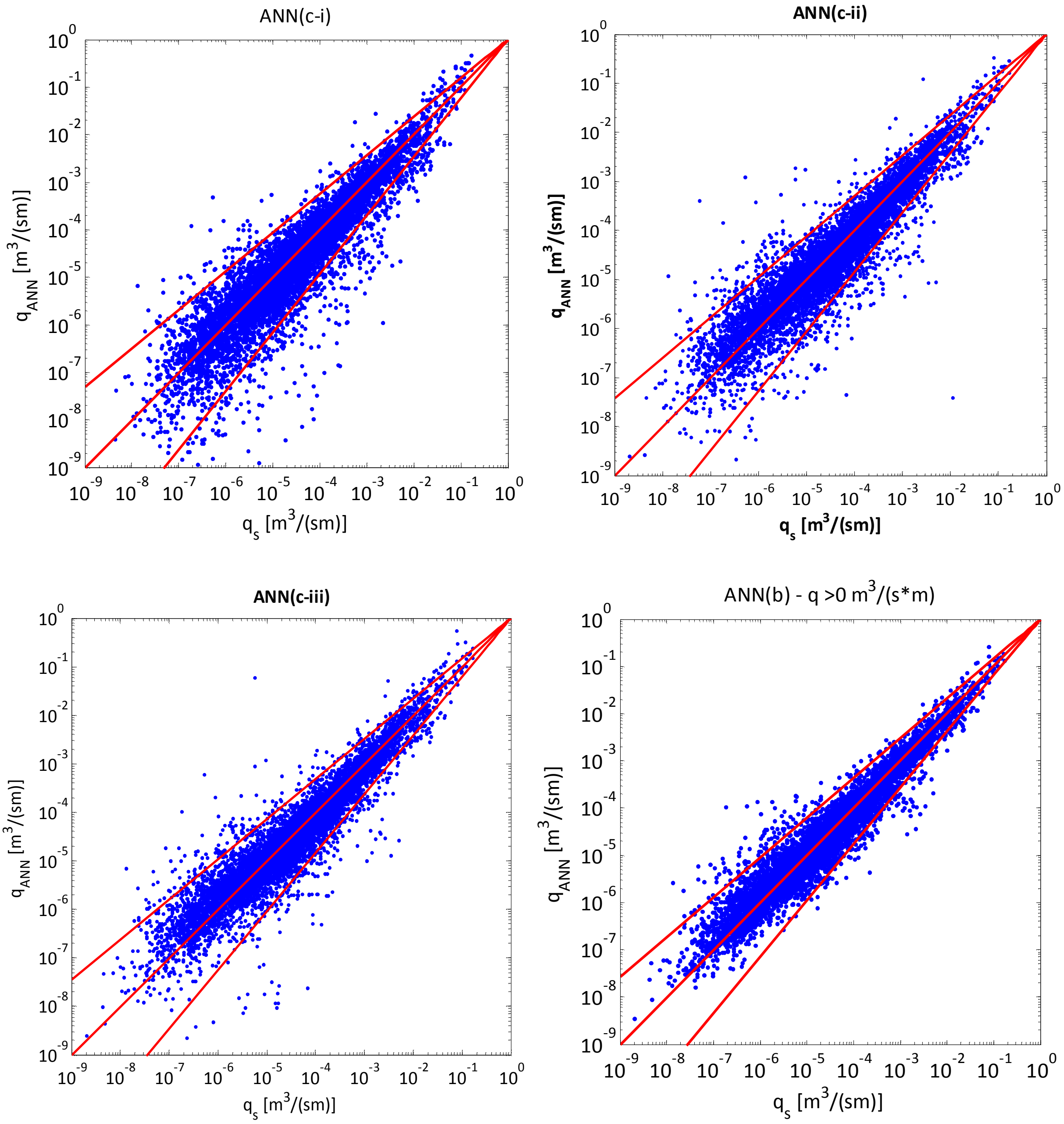


Figure 0.8 – ANNs predictions of the overtopping discharge (ordinate) vs the corresponding experimental values (abscissa). From left to right and from top to bottom, results relative to ANN (c-i), ANN (c-ii), ANN (c-iii), ANN (b). The three ANNs (c) have been trained on all the available overtopping tests reporting *RF* and *CF* ≠ 4, being the 0-overtopping values artificially replaced variously according to different methodologies. All the plots refer to the prediction of only q>0 data.

6.4 Quantifier-classifier-quantifier

This section aims to describe the finally developed model to predict both “large” and “small” values of *q*. It essentially consists in a two-phase ANN model composed by a first-phase classifier and two second-phase quantifiers. Each of the three ANNs composing the complete model adopts the optimized architecture presented in Chapter 4, but is trained on a different database.

The conceptual layout of such a model is sketched in Figure 6.9. The first phase ANN, working as a “quantifier-classifier”, provides a numerical prediction of the *q*-values (differently from Verhaeghe et al. (2008) classifier ANN, which produced a Boolean-type output). A second and a third ANN work in series after the classifier, and alternatively each other. According to the classifier predictions, these two ANNs respectively process the "non-significant classified" (Quantifier 1, in Fig. 6.9) and the "significant classified" (Quantifier 2, in Fig. 6.9) overtopping tests.

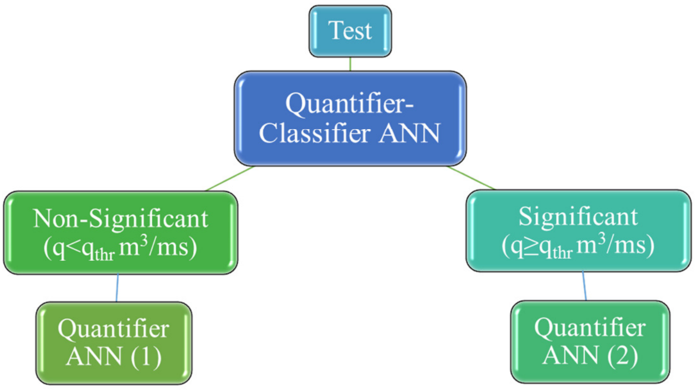

Test
Quantifier-Classifier ANN
Non-Significant ($q<q_{thr}$ m^3/ms)
Significant ($q\geq q_{thr}$ m^3/ms)
Quantifier ANN (1)
Quantifier ANN (2)

Figure 0.9 – Conceptual layout of the two-phase classifier-quantifiers optimized ANN model for the prediction of both "large" and "small" values of the wave overtopping discharge.

The two-phase methodology developed within this work aims to overcome the main failing attributed to the existing two-phase ANN (Verhaeghe et al. (2008)), i.e. the extension of the zero data based on the incorrect assumption that all the values set to zeros in CLASH database are really "zeros", while (see Paragraph 6.3) the definition of zero varied according to the authors who carried out the experiments.

This shortcoming is primarily overwhelmed through the assessment of the "uncertainty" connected to the zero values described in Paragraph 6.2.2 and the discard of those tests considered "unreliable". Secondly, the adoption of two distinct quantifiers in the second-phase ensures that no test pre-classified as "non-significant" is definitely rejected, but passed to a quantifier ANN.

A first subsection (Paragraph 6.4.1) briefly presents the steps of the developed methodology, while the following Paragraphs 6.4.2 and 6.4.3 are dedicated to the definition of the "optimized" ANNs to be adopted as classifier and quantifiers. The last Paragraph 6.4. illustrates the definitive architecture of the model and discusses its overall performance and behavior, both regarding the prediction of "large" and "small" *q*-values.

6.4.1 The steps of the methodology

So far, three differently trained ANNs have been presented, namely ANN (a), ANN (b) and ANN (c). These ANNs and the corresponding characterizing training databases are synthetized in Tab. 6.8. In particular, it is worthy to remark that ANN (c) is trained on a modified database including the original zero values conveniently "replaced" (see Fig. 6.6).

Table 0.8 – Synthesis of the three differently trained ANNs and the corresponding training database.

ANN	Training database (#)
a	$q\geq10^{-6}$ m^3/(sm) (# 7'716)
b	$q>0$ m^3/(sm) (# 8'783)

c	All q, with modified zeros (#9’896)

ANN (a) and ANN (b) resulted to be inadequate for the prediction of the “small” values of wave overtopping, while each ANN (c) cannot represent the “large” values with sufficient accuracy. Therefore, the main idea to solve the matter is to split the data to be predicted into two (or more) database and contemporarily employ two (or more) ANNs, specifically trained for the quantification of the different categories of data.

Such idea mainly requires:

- the definition of a classifier criterion (preferably, an ANN) to correctly split the database;
- the choice of two ANNs for the distinct prediction of the data.

Do the so far presented ANNs, or at least one of them, fit the role of a classifier or a quantifier?

In particular, the selection of a “reliable” and “robust” classifier is paramount, because of the error propagation from the first phase to the following one. In other words, if the classifier criterion gets a wrong prediction and passes a test to the wrong quantifier, the final prediction is going to be affected by an increase of the error and a general worsening of the first evaluation. Such effects would compromise the general efficiency of the model. Furthermore, which is the “optimal” threshold to be adopted in order to split the data? Does the “commonly” accepted value q = 10^{-6} $m^3/(sm)$ represent the best choice? This and other aspects will be analyzed in Paragraph 6.4.2.

After the classifier, defining an appropriate quantifier ANN is of course fundamental as well. An important matter, for example, is the choice of the most suitable database for the training of the quantifier ANNs. Is a narrower database preferable, in order to maximize the performance of the quantifiers on specific ranges of training, or is it better to choose a wider database, to improve the capability of generalization and minimize the error in case of a wrong selection from the classifier? Paragraph 6.4.3 shows the analysis of sensitivity carried out to answer these questions and provide the “optimized” quantifiers.

6.4.2 Definition of the quantifier-classifier

Principally, the definition of a classifier criterion involves the assessment of two issues:

1) The selection of the criterion itself: is an ANN sufficient to perform the selection or has it to be “supported” by any other model? Which is the “best” ANN to be used as a quantifier-classifier?
2) The choice of the threshold value to select the tests.

The first requirement that a classifier criterion must satisfy is the capability of generalization and therefore a “wide-enough” validity range. Indeed, the classifier is the first filter that must be able to deal with whatever input set a possible user would provide. Then, the achievement of an optimized performance is not the prime necessity, even if the error must be as lower as possible, in order to deliver the tests to the “correct” quantifier.

Starting from this consideration, the trained ANN (a) – being trained only against tests with q >10^{-6} $m^3/(sm)$, see Tab. 6.8 – does not clearly satisfy the requirement of “wideness” of the training range and for this reason has been discarded as a classifier. At the contrary, ANN (b) and ANN (c) potentially fit the requisite. Among the three ANNs (c), the one most suitable to be employed as a classifier is ANN (c-iii) (see Paragraph 6.3.2), since it provides the best "overall" predictions (see Tab. 6.7 and Fig. 6.8).

Then, both ANN (b) and ANN (c-iii) were tested as classifiers. The schemes of Fig. 6.10 show the logical layout of the two tested combinations: in these diagrams, the threshold value – not yet defined – is simply indicated as "q_{thr}" as well as the quantifier ANNs, which for the same reason are not specified.

Besides, in order to support the ANN predictions with the predictions performed by van der Meer et al. (2013) formula (see Eq. 6.2), a variation to these schemes was investigated. This way, a "double" criterion for the selection of the data, as it is illustrated in the outlines of Fig. 6.11, was introduced: if ANN and formula agree, the data are simply directed to the matching quantifier, according to q_{thr}; otherwise, the data are in any case passed to Quantifier 2 ("large" values). Since ANN (c-iii) is already influenced by the results of Eq. 6.2 (see Paragraph 6.3.2), this variation is expected to be more relevant for the application with ANN (b).

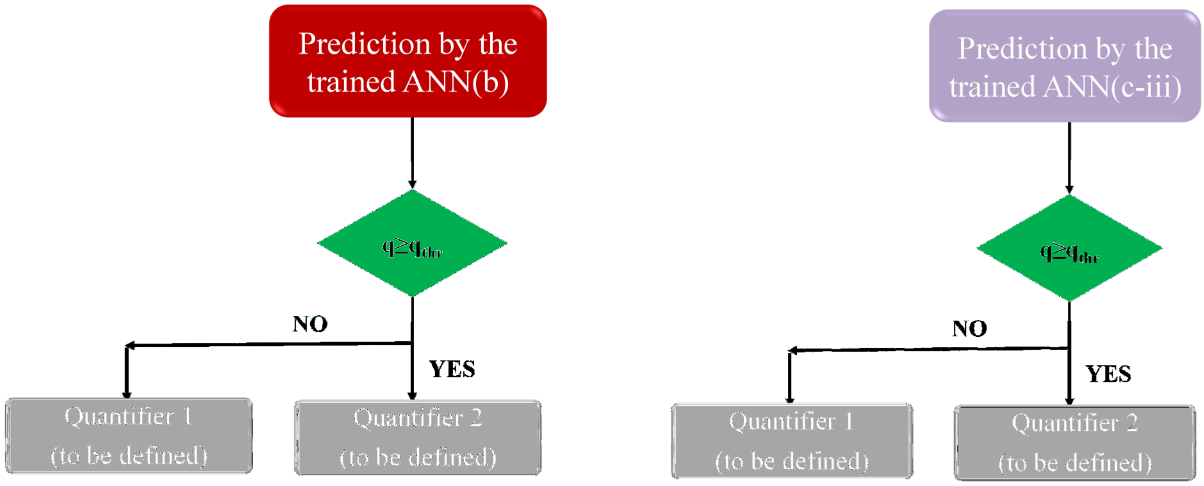

Prediction by the trained ANN(b)
q≥q_thr
NO
YES
Quantifier 1 (to be defined)
Quantifier 2 (to be defined)
Prediction by the trained ANN(c-iii)
q≥q_thr
NO
YES
Quantifier 1 (to be defined)
Quantifier 2 (to be defined)

Figure 0.10 – Logical layout and working principle of two tested combinations for the definition of the classifier criterion. Schemes adopting either ANN (b) (panel to the left) or ANN (c-iii) (panel to the right) as classifiers.

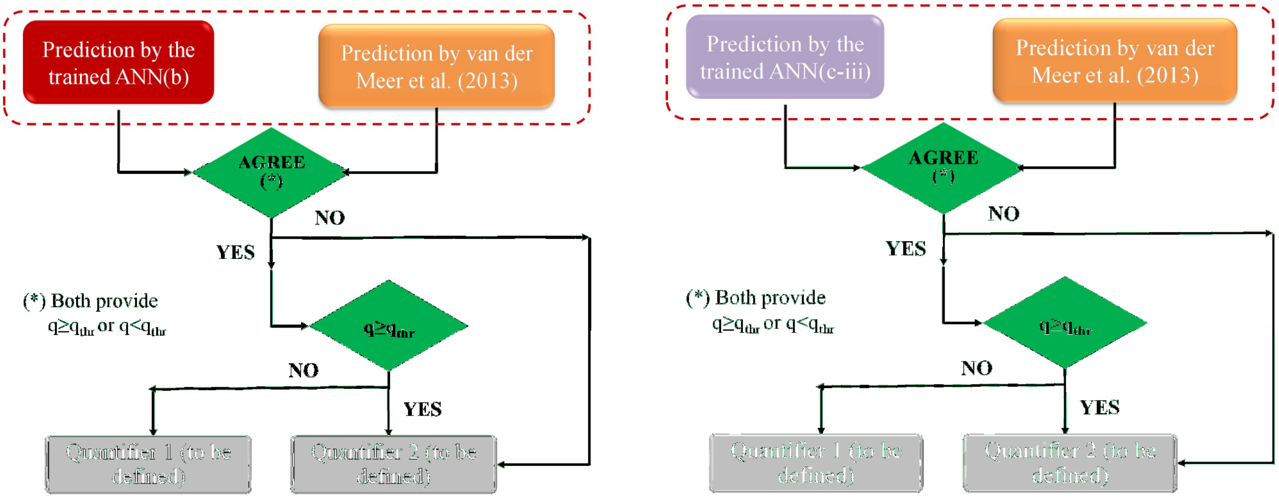

Prediction by the trained ANN(b)
Prediction by van der Meer et al. (2013)
AGREE (*)
NO
YES
(*) Both provide q≥q_thr or q<q_thr
q≥q_thr
NO
YES
Quantifier 1 (to be defined)
Quantifier 2 (to be defined)
Prediction by the trained ANN(c-iii)
Prediction by van der Meer et al. (2013)
AGREE (*)
NO
YES
(*) Both provide q≥q_thr or q<q_thr
q≥q_thr
NO
YES
Quantifier 1 (to be defined)
Quantifier 2 (to be defined)

Figure 0.11 – Logical layout and working principle of two tested combinations for the definition of the classifier criterion. The ANN classifier is supported by the introduction of van der Meer et al. (2013) formula. Schemes adopting either ANN (b) (panel to the left) or ANN (c-iii) (panel to the right) as classifiers.

All the quantitative results derived from these combinations are reported in Tab. 6.9. The predicted data correspond to the 8’783 tests reporting q>0: all these tests are firstly processed by the classifier, and then subdivided into two parts; then, each quantifier handles the respective portion and produces the corresponding evaluations. In order to ease the interpretation of the different tested combinations, from Tab. 6.9 on, the adoption of the colors blue and green has the following meaning:

- the blue indicates, for each case, the modified parameter object of the sensitivity analysis to the classifier criterion;
- the green enhances the “better” results, where “better” accounts of all the error indexes and the standard deviations.

Tab. 6.9 shows the indexes of performance and the number of errors relative to the “final” outcome of the model, i.e. to the separated predictions provided by the two quantifiers jointed together. The quantifier ANNs are the “optimized” ones as resulted from the ad-hoc sensitivity analysis presented in in the next Paragraph 6.4.3.

Table 0.9 – Synthesis and results of the tested combinations for the definition of the “optimal” classifier criterion. The predicting database includes all the non-zero overtopping tests (*q* > 0, 8’783 data). The inclusion/exclusion of the weight factors (WF) only regards the databases on which the applied ANNs were trained. The blue colour indicates for each case the modified parameter, while the green colour enhances the “better” case.

Prediction of *q*>0, (#8'783) – Selection of the classifier criterion						
Classifier ANN/Criterion	Threshold value $m^3/(sm)$	*WF*	*RMSE*	*WI*	R^2	# Large errors (%)
c(iii)	10^{-6}	no	0.058± 0.007	0.92 ±0.01	0.62±0.2	176 (2.0%)
c(iii)	10^{-7}	no	0.052± 0.005	0.964 ±0.009	0.86±0.2	437 (5.1%)
c(iii) + VDM pred.s	10^{-7}	no	0.049± 0.004	0.965 ±0.009	0.87±0.1	261 (3.0%)
c(iii) + VDM pred.s	10^{-7}	yes	0.049± 0.004	0.967 ±0.008	0.87±0.09	267 (3.0%)
b	10^{-6}	no	0.053± 0.007	0.94 ±0.01	0.73±0.1	188 (3.0%)
b	10^{-7}	no	0.048± 0.005	0.975 ±0.008	0.91±0.09	268 (3.0%)
b + VDM pred.s	10^{-7}	no	0.046± 0.005	0.974 ±0.009	0.90±0.1	158 (1.8%)

It is worthy to remark that the definition of the complete classifier-quantifiers model of course required a simultaneous sensitivity analysis to the classifier criterion and the quantifier ANNs. However, in this work it has been preferred to separate the analysis into two steps for clarity and simplicity reasons.

The individuation of the “better” results was driven by accounting more of the values of *WI* and R^2 and the corresponding standard deviations than *rmse* and the number of large errors. These indexes, being non-dimensional and therefore independent by the order of magnitude of the *q*-values, are considered more representative of the overall performance. At the contrary, *rmse* strictly depends – by definition – on the measure of *q*, and therefore, the lower the values of *q*, the lower the *rmse*. This issue does not affect the comparison among different applications if the predicting database is always the same. When instead, a classifier is defined and the overall data are differently distributed to the two quantifiers, the numerical *rmse* is affected in dependence of the different proportion of data delivered to each quantifier at each different combination.

Following this approach, i.e. by contemporarily accounting of the classifier and the quantifiers' performance and privileging the values of *WI* and R^2, the resulting "better" combination for the definition of a classifier criterion matches to the adoption of:

- ANN (b) as classifier ANN, i.e. ANN model trained on all non-zero data;
- exclusion of the *WF* from the ANN training process;
- no introduction of van der Meer et al. (2013) formula to validate ANN predictions;
- threshold value: $q_{thr} = 10^{-7}$ m^3/(sm).

The combination, which accomplishes all these requisites, is the one enhanced in Tab. 6.9 using the green color. The relative scheme of such a model is sketched in Fig. 6.12.

By comparing the performance associated to the "best" combination of Tab. 6.9 to the one of Tab. 6.1, relative to the simply quantifier ANN (b), only a modest decrease of *WI* (0.975 instead that 0.977) and *R* (0.91 instead of 0.92) is detected. This result, if confirmed by an improvement of prediction of zero values and a "qualitative" reduction of the overestimation bias, could lead to the conclusion that the adoption of a two-phase classifier-quantifiers overcomes the limits of the simply quantifier ANN.

These aspects are discussed in the following Paragraph 6.4.3.

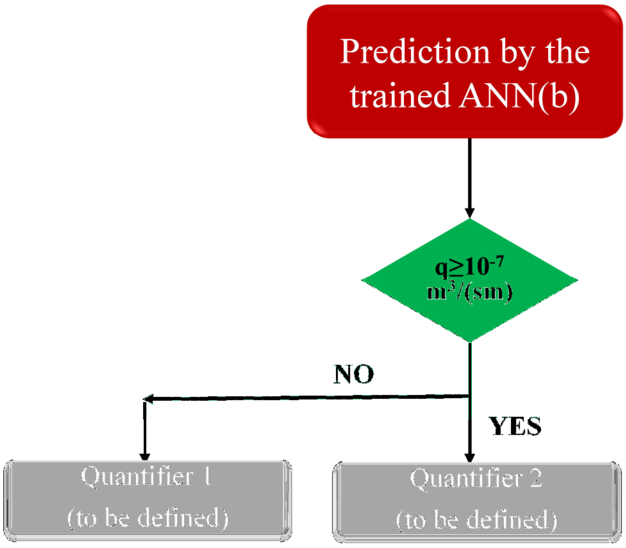

Prediction by the trained ANN(b)
q≥10-7 m3/(sm)
NO
YES
Quantifier 1 (to be defined)
Quantifier 2 (to be defined)

Figure 0.12 – Logical layout and working principle of the optimized classifier criterion. The scheme adopts the trained ANN (b) and the value of 10^{-7} m^3/(sm) as threshold to distribute the data between the two quantifiers (to be defined yet).

6.4.3 Definition of Quantifier 1 and Quantifier 2

Once defined the classifier ANN, i.e. ANN (b), and the threshold criterion, i.e. 10^{-7} m^3/(sm), the final step consisted in the selection of the two quantifiers for the prediction of, respectively, small ("Quantifier 1") and large ("Quantifier 2") values of wave overtopping discharge.

Besides the already presented ANNs – ANN (a), ANN (b) and the three ANNs (c) – the definition of Quantifier 1 and 2 has required the creation of other ANNs, differing from the existing ones for the training database. Mainly, two kinds of ANNs have been developed:

- ANN (d), trained on a database of 8’619 tests of $q > 10^{-7}$ $m^3/(sm)$; this ANN – whose training database represents a “compromise” between ANN (a) and ANN (b) – could fit the Quantifier 2;
- ANN (e), to fit the Quantifier 1, trained on a size-variable database including only test with $q \leq 10^{-5}$ $m^3/(sm)$; the size of the database depends on the inclusion or not of the zero tests and the way they have been “replaced” (see Paragraph 6.3.2). According to the different training database, four variations of ANN (e) were defined. The schemes of Fig. 6.13 aim to synthetically describe the methodological steps followed to produce each ANN (e) and characterize the corresponding training database.

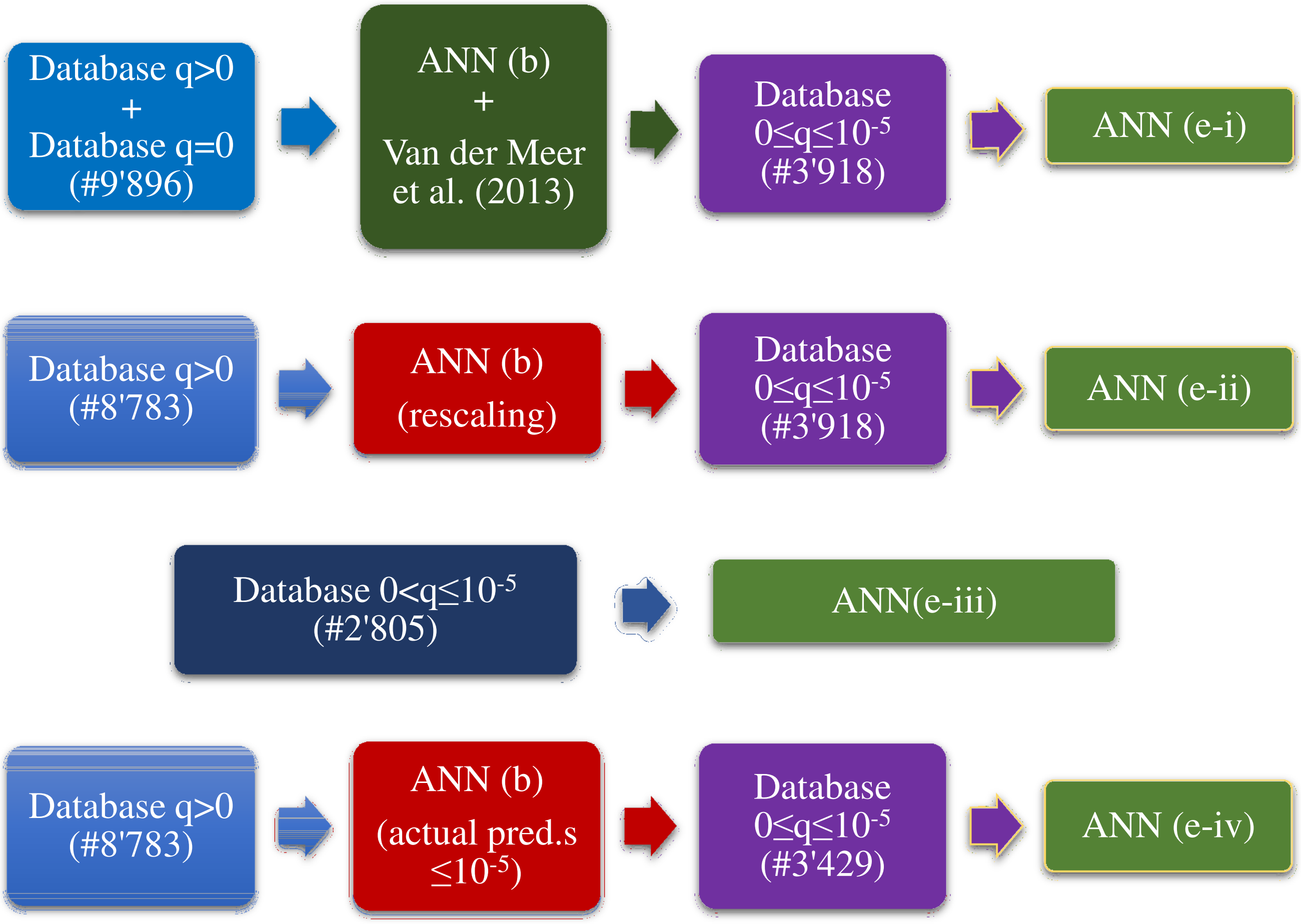

Database q>0 + Database q=0 (#9'896)
ANN (b) + Van der Meer et al. (2013)
Database 0≤q≤10-5 (#3'918)
ANN (e-i)
Database q>0 (#8'783)
ANN (b) (rescaling)
Database 0≤q≤10-5 (#3'918)
ANN (e-ii)
Database 0<q≤10-5 (#2'805)
ANN(e-iii)
Database q>0 (#8'783)
ANN (b) (actual pred.s ≤10-5)
Database 0≤q≤10-5 (#3'429)
ANN (e-iv)

Figure 0.13 – Schemes illustrating the several variations to ANN(e), trained on a size-variable database including only test with $q \leq 10^{-5}$ $m^3/(sm)$. Each scheme reports the methodological steps followed to produce the corresponding ANN.

Table 6.10 illustrates the main features of ANN (d) and the several variations to ANN (e). The choice to train each ANN (e) up to 10^{-5} (being 10^{-7} the threshold values which subdivides the data between the two quantifiers) is due to the necessity of partial overlapping of the range of validity of the two quantifiers. Indeed, in case of a classifier wrong prediction occurs, it is essential that the receiving quantifier is able to deal with an out-of-range input test.

Table 0.10 – Synthesis of the ANNs and the corresponding training database appositely created for the definition of the two quantifier ANNs.

ANN	Training database (#)	Technique of zeros replacement	Quantifier
d	$q \geq 10^{-7}$ $m^3/(sm)$ (# 8'619)	-	2 – "large"
e(i)	$0 \leq q \leq 10^{-5}$ $m^3/(sm)$ (#3'918)	Predictions of zeros by ANN (b) and VDM formula	1 – "small"
e(ii)	$0 \leq q \leq 10^{-5}$ $m^3/(sm)$ (# 3'918)	Rescaling of the predictions of zeros by ANN (b)	1 – "small"
e(iii)	$0 < q \leq 10^{-5}$ $m^3/(sm)$ (# 2'805)	-	1 – "small"
e(iv)	$0 \leq q \leq 10^{-5}$ $m^3/(sm)$ (# 3'429)	Predictions of zeros by ANN (b) resulting $<10^{-5}$	1 – "small"

Similarly, to the analysis of performance carried out for the selection of the "optimized" classifier, many different combinations of Quantifier 1and Quantifier 2 were verified. Because of the defined ANNs, the schemes of Fig. 6.14 show the several tested possibilities to define the two Quantifiers.

Indeed, the choice of the "optimal" quantifiers should contemporarily account for all the following aspects:

1) The "overall" performance of the two-phase model, i.e. the global outcome provided by the sequence of classifier and quantifiers;
2) The performance related to the predictions of the "significant" values of q: it is essential to ensure that the new model does not provide any worsening respect to the "simply" quantifier ANN trained against only the data $q>10^{-6}$ $m^3/(sm)$;
3) The performance related to the predictions of the zero-values of q, since the two-phase model was conceived to improve the overestimation of these values.

For this purpose, the performance of the different combinations of the classifier-quantifiers model was evaluated against the prediction of three different databases. The derived results are gathered within the Tables 6.11, 6.12 and 6.13, which respectively refer to a predicting database of: 8'783 tests $q > 0$; 7'716 tests $q \geq 10^{-6}$ $m^3/(sm)$ ("significant" overtopping); 1'113 zero-overtopping tests. In these tables, the blue colour indicates the specific case of ANN analyzed and the green aims to heighten the best performance.

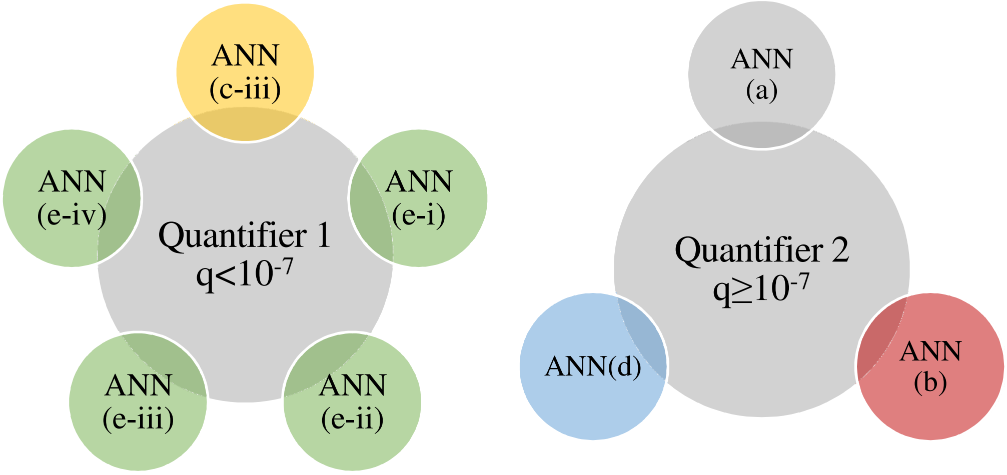


Figure 0.14 – Schemes reporting all the ANNs to be employed as Quantifier 1 for the "small" overtopping tests (to the left) and as Quantifier 2 for the "large" overtopping tests (to the right).

Within Tab. 6.12 one case of combination that still adopts the "double" classifier (i.e. ANN (b) and Eq. 7 predictions) is present: this test has been performed in order to confirm and enforce the reason which has leaded to the exclusion of van der Meer et. (2013) "support" from the classifier. Indeed, from Tab.6.9, this combination shows just a little worsening of the overall performance, while from Tab. 6.12, the failing of this model against the prediction of “significant” wave overtopping is evident.

Table 0.11 – Synthesis and results of the tested combinations for the definition of the “optimal” quantifier ANNs. The predicting database includes all the experimental non-zero overtopping tests (q>0, 8’783 data). The blue colour indicates for each case the modified parameter, while the green colour enhances the “better” case.

Prediction of *q*>0, (#8'783) – Selection of the quantifiers							
Classifier ANN	Quantifier 2 - “large”	Quantifier 1 - “small”	*WF*	*RMSE*	*WI*	R^2	# Large errors (%)
b	b	e(i)	no	0.048± 0.005	0.975 ±0.008	0.91±0.09	268 (3.0%)
b	b	c	no	0.048± 0.005	0.975 ±0.008	0.91±0.03	268 (3.0%)
b	b	e(ii)	no	0.049± 0.005	0.976 ±0.008	0.91±0.06	268 (3.0%)
b	b	e(iii)	no	0.049± 0.005	0.976 ±0.008	0.91±0.04	268 (3.0%)
b	b	e(iv)	no	0.049± 0.005	0.976 ±0.008	0.91±0.03	268 (3.0%)
b	a	e(ii)	no	0.064± 0.004	0.964 ±0.006	0.87±0.05	918 (10.4%)
b	d	e(ii)	no	0.053± 0.005	0.974 ±0.007	0.90±0.05	484 (5.5%)

By comparing the results reported in Tab. 6.11, it is evident that the best performance as Quantifier 2 is provided by ANN (b). ANN (a) presents values of the error indexes and a number of “large errors” non-comparable to ANN (b), and for this reason it was discarded. The *ad-hoc* trained ANN (d) revealed to work better than ANN (a), but not as much as ANN (b), producing a double number of large errors and lower values of both *WI* and R^2.

Table 6.12, focused on the prediction of the “significant” wave overtopping, definitely settles ANN (b) as the Quantifier 2. Its performance is substantially identical to both ANN (d), and ANN (a), which – differently from ANN (b) – is there applied to the prediction against its own training database. The *rmse* index and the number of large errors – which for this application case of essential coincidence of predicting databases among the different combinations – are significantly lower for ANN (b), compensating the slight higher values of *WI* and R^2 associated to ANN(d) and ANN (a).

Table 0.12 – Synthesis and results of the tested combinations for the definition of the “optimal” quantifier ANNs. The predicting database includes all the experimental overtopping tests reporting q>10^{-6} m^3/(sm) (7’716 data). The blue colour indicates for each case the modified parameter, while the green colour enhances the “better” case.

Prediction of *q*>10^{-6} m^3/(sm) (#7'716) – Selection of the quantifiers							
Classifier ANN	Quantifier 2 - “large”	Quantifier 1 - “small”	*WF*	*RMSE*	*WI*	R^2	# Large errors (%)
b	b	e(i)	no	0.043± 0.004	0.975± 0.007	0.91± 0.03	51 (0.7%)
b + VDM pred.s	b	e(i)	no	0.039± 0.005	0.94± 0.02	0.72± 1	-
b	b	c	no	0.043± 0.004	0.975± 0.007	0.91± 0.03	51 (0.7%)
b	b	e(ii)	no	0.043± 0.004	0.975± 0.007	0.91± 0.03	51 (0.7%)
b	b	e(iii)	no	0.043± 0.004	0.975± 0.007	0.91± 0.03	51 (0.7%)
b	b	e(iv)	no	0.043± 0.004	0.975± 0.007	0.91± 0.03	51 (0.7%)
b	a	e(ii)	no	0.045± 0.003	0.978 ±0.004	0.92±0.01	150 (1.9%)
b	d	e(ii)	no	0.046± 0.004	0.976 ±0.006	0.91±0.02	94 (1.2%)

The plots of Fig. 6.15 qualitatively compare the predictions of the original one-phase quantifier ANN (a) to the ones of ANN (b) employed as quantifier in the two-phase model. The overestimation error concentrated within 10^{-5}÷10^{-6} in the diagram of ANN (a) (which is the same of Fig. 6.2, to the left) is sensibly reduced by ANN (b), and the confidence bands are visibly narrower for ANN (b), according to the lower values of *rmse*.

It can be concluded that the two-phase classifier-quantifier ANN (b) improves the prediction of the "significant" overtopping with respect to the one-phase ANN (a) and, without compromising the overall performance, partially solves the shortcoming of the biased overestimation of values within 10^{-5}÷10^{-6}.

The superior precision and robustness of predictions of ANN (b) definitely demonstrates as the wideness of the training ranges and the generalization capability are the most important requisite that an ANN model should accomplish, no matter if working as a classifier or a quantifier. This concept is furtherly confirmed by the improved performance associated to ANN (d) – trained on 8'619 tests of $q > 10^{-7}$ – with respect to ANN (a) – trained on 7'716 tests of $q > 10^{-6}$.

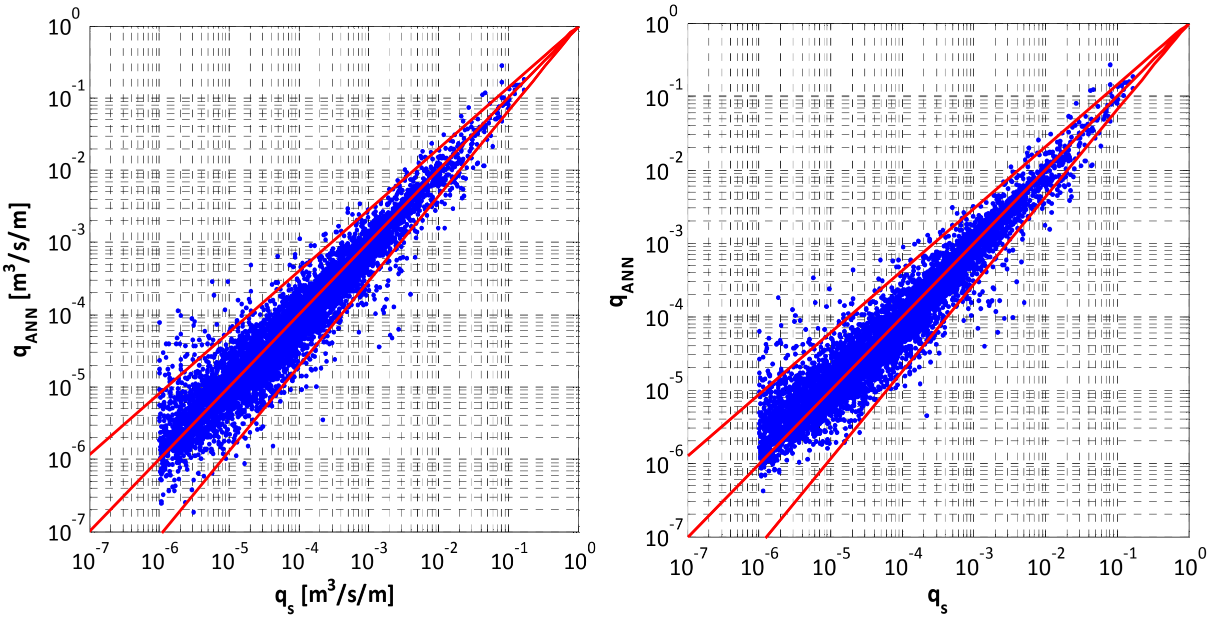


Figure 0.15 – ANNs predictions of the overtopping discharge (ordinate) vs the corresponding experimental values (abscissa). Predictions of the experimental values of $q \geq 10^{-6}$ obtained from the optimized classifier-quantifiers (to the left) and from the original simply quantifier ANN trained on values of $q \geq 10^{-6}$ (to the right).

As regards the choice of Quantifier 1, from Tab. 6.11 no conclusion can be achieved. All the results relative to the employment of ANN (b) as Quantifier 2 lead essentially to a same overall performance, due to the larger amount of data passed to Quantifier 2 (instead that Quantifier 1) and the consequent higher weight attributed to its predictions.

Tab. 6.12 even does not show any difference among the several ANNs employed as Quantifier 1. This can be explained by considering the prediction database ($q \geq 10^{-6}$ tests only) and the threshold for the repartition of data ($q_{thr}=10^{-7}$): all the input data are passed to Quantifier 2 and none to Quantifier 1, a further important confirmation of the good behavior of ANN (b) as a classifier. Actually, the classifier underestimates no input value of more than one order of magnitude.

The "optimal" Quantifier 1 is instead indicated in Tab. 6.13, from the comparison among the prediction of zeros. From this table, it is evident that only ANN (e-ii) can accomplish the role of quantifier of the "small" and zero values. The number of wrong predictions (only nine values out of 1’113 are predicted greater than 10^{-6} and none greater than 10^{-4}) is so much lower than the other applications that no further comment is required.

Table 0.13 – Synthesis and results of the tested combinations for the definition of the “optimal” quantifier ANNs. The predicting database includes the experimental zero-overtopping tests (q=0, 1’113 data). The blue color indicates for each case the modified parameter, while the green color enhances the “better” case.

Prediction of *q* = 0 (#1'113) – Selection of the quantifiers					
Classifier ANN	Quantifier 2 - “large”	Quantifier 1 - “small”	*WF*	Wrong (#>10^{-6})	Wrong (#>10^{-4})
b	B	e(i)	no	168 (15%)	19 (2%)
b	B	c	no	249 (22%)	45 (4%)
b	B	e(ii)	no	9 (0.8%)	0 (0%)
b	B	e(iii)	no	470 (42%)	56 (5%)
b	B	e(iv)	no	464 (42%)	49 (4%)
b	A	e(ii)	no	9 (0.8%)	0 (0%)
b	D	e(ii)	no	9 (0.8%)	0 (0%)

Therefore, the final outline of the two-phase classifier-quantifiers purposed in this work adopts:

- ANN (b) as the first-phase classifier ANN;
- threshold value for the partition of the data between the two quantifiers: $q_{thr} = 10^{-7}$ m^3/(sm);
- ANN (b) as the second-phase quantifier for the large-classified values of *q*;
- ANN(e-ii) as the second-phase quantifier for the small-classified values of *q*.

The qualitative performance of the optimized model is presented in Fig. 6.16, which compares the model predictions to the experimental non-zero values (8’783 tests of q>0). In this plot, the estimations of *q* provided by Quantifier 1 are displayed in green color, to be distinguished from the predictions provided by Quantifier 2 (blue color). In this figure, the Quantifier 1 is affected by an underestimation bias, which can be explained by considering that it was trained on artificially lowered values (the rescale ANN predictions of zeros).

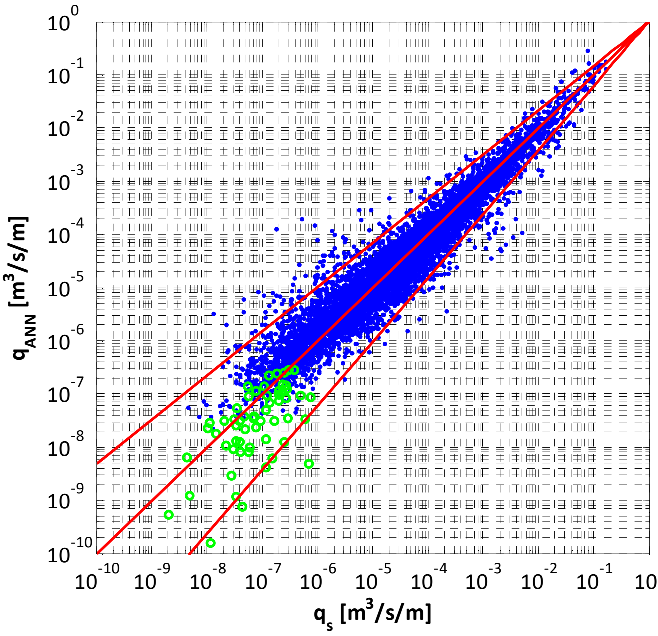


Figure 0.16 – ANNs predictions of the overtopping discharge (ordinate) vs the corresponding experimental values (abscissa) obtained from the optimized two-phase classifier-quantifiers. Predicting database of 8’783 tests of q>0. The blue captions refer to the predictions provided by the quantifier for the “large” overtopping values, while the green captions refer to the predictions provided by the quantifier for the “small” values.

igure 6.17 finally portrays the optimized layout of the two-phase classifier-quantifiers ANN model. The adoption of a two-phase model is required, even if the classifier ANN coincides with the Quantifier 2, in order to avoid the bias of overestimation of small and zero values. Indeed, the behavior of ANN (b) against the prediction of the zeros has been already discussed in Paragraph 6.2.2 and the percentage of wrong predictions of zeros is extremely higher (44% instead of 1%, see Tab. 6.2 and Tab. 6.13).

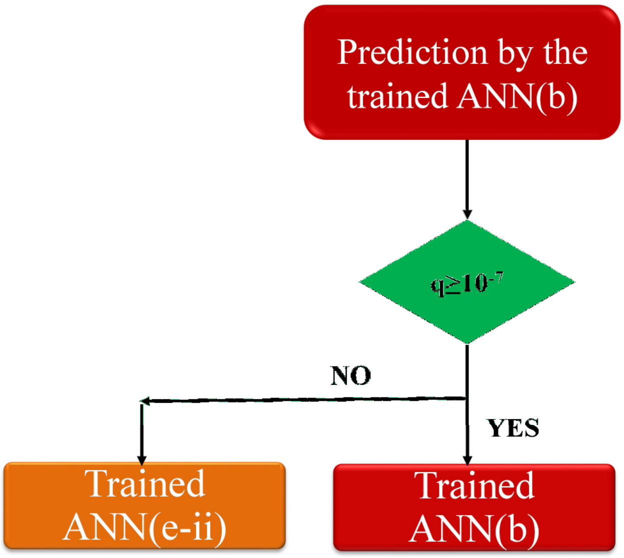


Figure 0.17 – Logical layout and working principle of the optimized two-phase classifier-quantifiers ANN model. The scheme adopts the trained ANN (b) as a classifier, the value of 10^{-7} m^3/(sm) as threshold to distribute the data between the two quantifiers, again ANN (b) as quantifier for the large-classified tests and ANN(e-ii) as a quantifier for the small-classified tests. ANN (b) is trained on 8’783 tests of q>0 and ANN(e-ii) on 3’918 tests of 0≤q≤10^{-5} m^3/(sm).

6.5 Conclusion about the two-phase classifier-quantifiers ANN

The architecture of the ANN has been revised and now includes a classifier/quantifier and two quantifiers. The final proposed structure of such two-phase model adopts the same ANN for the first-phase classifier/quantifier and for the second-phase quantifier for the prediction of the “large” values of q. Such ANN was trained on the whole database for q>0 (being excluded all the tests reporting either RF=4 or CF=4). The quantifier for the “small” values is an ANN trained on the range $0 \leq q \leq 10^{-5}$ m^3/(sm), with the inclusion of the “zero values”.

The “zero values” in the database are substituted by the prediction of the ANN and rescaled to < 10^{-6} m^3/(sm), in order to avoid the bias in the prediction due to the overestimation of the small values of q.

This revised classifier-quantifiers leads to accurate and improved quantitative predictions for $q \geq 10^{-6}$ m^3/(sm), and good predictions of the cases with q <10^{-6}. It is able to deal with the “zero values”, providing less than 1% of overestimations.

The sensitivity analysis to the differently tested ANNs allows concluding that one of the paramount requirement for the proper employment of an ANN is the wideness of the training database. Actually, the classifier and the quantifier for the “large” values correspond to the

ANN trained on the widest range of data, with the exclusion of the “zero values”, since these tests would lead to an under-estimation bias if used to train a unique ANN.

However, the adoption of a two-phase ANN complicates the model architecture and the artificial replacement of the zeros could represent a weak point. Therefore, an extension of the database with data about “small” values of q would be desirable. Such data would be employed to properly train the ANN with “real” and “reliable” tests. In conclusion, the research to optimize the representation of the wave overtopping is still in progress.

7. CONCLUSIONS

The steps and the main outcomes of this research are hereinafter synthesized. A subdivision in sections according to the different topics is employed.

7.1 The database

The first and paramount requirement to develop a “good” Artificial Neural Network (ANN) is the availability of an experimental database for the calibration and the validation of the model. Therefore, the first step of this work consisted in the arrangement of a “wide-enough” and homogeneous collection of tests, organized following the same structure schematization.

Starting from the original wave overtopping database gathered within the project CLASH (2004), a new extended database was prepared, following the same structural and hydraulic schematization as in CLASH. The following additional information were included, where available: the wave transmission and wave reflection coefficients K_t and K_r, and the average unit size D as representative of the structure elements.

The final database is composed by 287 experimental datasets, organized into 7 “sections”, labeled progressively from A to G, in order to distinguish the different type of structures and wave attack conditions: rock permeable straight slopes (group “A”), rock impermeable straight slopes (group “B”), armour units straight slopes (“C”), smooth and straight slopes (“D”), structures with combined slopes and berms (“E”), seawalls (“F”) and oblique wave attacks (“G”).

Within the overall database, the data are so distributed: 11’825 tests on the wave overtopping discharge q, 7’413 tests on K_r and 3’366 on K_t; among these test, 2’065 include both q and K_r; 2’303 include both K_r and K_t; less than 100 tests include all the three parameters q, K_r and K_t.

Each test within the new database is described by the same 36 parameters, of which:

- 14 hydraulic parameters, characterizing the wave attack conditions;
- 18 structural parameters, for the as general as possible description of the cross-section of the structures;
- 4 general parameters, the reliability and the complexity factors, the identify label of the test and the identifying marks of the armour unit/type.

7.2 The new ANN

The choice of the best ANN layout has represented the second step of the work. The overall ANN “layout” accounts of both the input parameters and the internal architecture.

The definition of the ANN layout has consisted in a step-by-step optimization process based on a progressive work of revision of the parameters, according to the results obtained by comparing the model outcomes to the existing ANNs and formulae. The process has leaded to the realization of a single ANN model able to predict all the three wave-interaction phenomena introduced above.

The following subsections briefly synthesizes the main features of the resulting input parameters and of the ANN architecture.

7.2.1 The ANN input parameters

The optimized input set is inspired by two of the existing ANNs – the CLASH ANN (2004) and the ANN proposed by Zanuttigh et al. (2013) – through a work of analysis, comparison and revision of their input layouts. It definitely comprehends 15 non-dimensional input elements, whose definition involves 16 dimensional parameters of the database. Each input element represents a specific physical parameter or process and describes a characteristic of the wave attack or of the structure geometry.

The wave attack input elements represent the most relevant physical processes of wave transformation, i.e.:

- wave breaking related to wave steepness;
- wave breaking related to water depth;
- shoaling;
- wave obliquity;

The geometrical input elements include:

- the downstream slope, which is the most relevant for reflection;
- the "average" slope in the run-up/run-down area, the most relevant for overtopping and transmission;
- the foreshore slope to account for shoaling effects;
- berm width and submergence (to describe the berm breakwaters);
- toe width and submergence (in case the wave breaking occurs over the toe and to account for a phase-delay of the reflected waves);
- crest/crown wall height;
- nominal size representative of the armour units, as an indicator of the wave pressure within the pores of the structures.

All the characteristics widths are non-dimensionalized with the wave length, to account for the induced local reflection; all the characteristics heights are non-dimensionalized with the water depth, to account for the induced local breaking – in case of the crest / wall height, to account for the real overtopping.

The use of physically-based dimensionless parameters showed an improved prediction capability for all the three output q, K_r and K_t.

7.2.2 The ANN architecture

The optimal architecture of the new ANN was investigated by means of a careful sensitivity analysis to several characteristic elements which lead the training and the learning processes, such as the training algorithm, the number of hidden neurons and the techniques to improve the ANN generalization.

The resulting characteristics of the new ANN architecture are resumed in the following:

- multilayer network, based on a "feed-forward back-propagation" learning algorithm; 1 hidden layer, and 1 output neuron, corresponding either to K_r, K_t or q.
- the hidden layer comprehends 40 hidden neurons;
- training algorithm: *Levenberg – Marquardt*, belonging to the category of the high performance algorithms based on the feedforward-backpropagation, which speed the convergence process;

- hidden neurons transfer function: hyperbolic tangent sigmoid function;
- output neuron transfer function: linear transfer function.
- error type: *mse* (mean squared error);
- method to improve generalization: none, after testing (and deciding to discard) the "*early stopping*" method. The assessment of the ANN performance and its capability of generalization are attributed to the *bootstrap resampling* technique.
- maximum number of iterations (*epochs*) allowed: 100;
- adoption of the weight factors (*WF*) to drive the bootstrap resampling: no.

The employment of bootstrap resampling technique allows to improve the capability of generalization of the ANN, i.e. the capability to predict outputs beyond the range of the training parameters, besides assessing its performance. This strategy is called "commitment of networks" and was already adopted in previous applications, but it was never provided a comparison with the results obtained by applying and discarding it.

One of the most significant and exportable findings of this research is that the exclusion of the *WF* from the bootstrapping carries out an improvement of both ANN performance and generalization. The results of this sensitivity analysis confirmed the suppositions that the attribution of the reliability and complexity factor (*RF* and *CF*) to each test might be affected by a certain subjectivity, according to the laboratory or the team who performed the tests, especially for the definition of *RF*. Moreover, the low weighting of some very complex tests (reporting *CF*=3) might induce the opposite effect of worsening the overall ANN performance. Indeed, the more composite the cross-section of a structure, the increased necessity for the ANN to recognize that pattern and therefore the increased necessity to select that test during the training phase.

It is important to remark that all the tests reporting either *RF*=4 or *CF*=4 were anyway excluded from the training.

7.3 Results of the ANN

The ANN proposed in this wok proved to be able to accurately predict the overtopping discharge q, the wave transmission coefficient K_t and the wave reflection coefficient K_r for a wide range of (complicated) structure geometries and wave attacks.

The results of the ANN, with reference to the parameters q, K_r and K_t, were analyzed and the comparison among predicted and experimental values was carried out.

For the quantitative examination of the ANN predictions, three error indexes were employed: the root mean square error, *rmse*, the Willmott index, *WI* (Willmott, 1981*)* and the coefficient of determination, R^2. The adoption of different indexes is introduced to take into account different aspects of the ANN performance and to compare the results of the different applications.

All the numerical indexes correspond to the average results obtained from 50 bootstrap resamples of the database and to each index associated the corresponding value of the standard deviation. The distribution of the errors – revealing to follow a Gaussian curve – allowed the adoption of standard deviations and confidence intervals to assess the uncertainty associated to the predictions.

Besides, a definition of "large errors" was given, in reference to the percentage of tests (with respect to the total number of tests) for which the ANN had systematically (in more than the 50% of resamples) predicted a value of the output parameter which differs from the experimental value more than the 50%.

The computation of these performance indexes revealed that the predictions are satisfactorily accurate, providing values of the *rmse* in the range of [0.03; 0.05,], of *WI* always greater than 0.98 (and approximately equal to 1 for K_r and K_t). The percentages of large errors are lower than the 9% for each application, and is around 2% for q.

The qualitative analysis of the error distribution underlined the importance of the homogeneity and extension of the database.

The accuracy of the ANN predictions for each of the three parameters was compared to existing formulae available in literature (i.e.: van der Meer et al. (2013) for q; van der Meer et al. (2005) for K_t.; Zanuttigh and van der Meer (2008) for K_r).and existing ANNs already developed for each of the single process (i.e.: Van Gent et al. (2007) for q; Panizzo and Briganti (2007) for K_t.; Zanuttigh et al. (2013) for K_r).

The comparison with existing formulae showed that the new ANN provides sensibly more accurate predictions for all the three applications q, K_r and K_t. An important result is that the ANN overcomes the fields of validity and the intrinsic discontinuities of the traditional approaches. The improvement can be explained by considering that the ANN adopts a greater number of parameters and involves several non-linear relationships which allow the achievement of a more "complete" degree of interpretation of the physical phenomena.

As regards the comparison with existing ANNs, the better results are obtained within the application to K_t, which demonstrates an undoubted improved performance. The application to K_r is in any case satisfactory, since, despite a little increase of the scatter, the new ANN involves an extension of the validity ranges and its distribution of the error is characterized by higher degree of symmetry.

Concerning q, the new ANN predictions are approximately as accurate as the CLASH ANN ones, despite a reduced quantitative error. The scarce improvement may be ascribed to the extension of the database and the inclusion of "complex" structures.

Furthermore, the new ANN is partially affected by an overestimation of the small values of q. It was proved that this shortcoming is induced by the decision to discard the values of q lower than the *a-priori* fixed threshold q=10^{-6} $m^3/(sm)$. This suggestion was indicated by Van Gent et al. (2007), who classified these tests as "poor reliable".

Both these aspects (complex geometries and small values of q) constitute elements of further investigations, partially carried out within this research and partially postponed to further research.

7.4 Ongoing and further research

7.4.1 Prediction of small values of *q*

A final step of this work focused on the possibility to and solve the problem of overestimation bias of "small" values of q (i.e. $q<10^{-6}$ $m^3/(sm)$), and improve the ANN predictions of such values. In this sense, an attempt to develop a logic classifier – according to Verhaeghe et al. (2008) approach – to replace the arbitrary threshold and enlarge the field of validity of the overtopping ANN was investigated but discarded because of an observed worsening of the overall performance.

Though, the effects of introducing a different type of two-phase model were verified through a revision of the ANN architecture. The final proposed structure is composed by a first-phase classifier/quantifier which provides quantitative predictions of q and two separate second-phase quantifiers, which respectively process the “large” (q>10^{-6} $m^3/(sm)$) and the “small” (q<10^{-6} $m^3/(sm)$) classified values of q.

A same ANN constitutes both the first-phase classifier/quantifier and the second-phase quantifier for the prediction of the “large” values of q. Such ANN was trained on the whole database for q>0 (being excluded all the tests reporting either *RF*=4 or *CF*=4).

The quantifier for the “small” values is a different ANN, trained on the lowest values of q, i.e. within the range $q \leq 10^{-5}$ $m^3/(sm)$. It is worthy to remark that the ranges of training of the two ANNs were on purpose overlapped in order to ensure the continuity of the predictions in case of classifier’s errors (i.e. in case a test is delivered to the wrongly quantifier).

This revised classifier-quantifiers leads to accurate and improved quantitative predictions for $q \geq 10^{-6}$ $m^3/(sm)$, and good predictions of the cases with q <10^{-6} $m^3/(sm)$. It is also able to deal with the identically zero values of the database (never included in the training), providing less than 1% of overestimations.

Nevertheless, an enlargement of the overtopping dataset with new tests of modest or null values of q is wished to improve the ANN skill to reproduce small values of overtopping.

7.4.2 Representation of complex geometries

Both the applications to K_r and q revealed that the greatest and most recurrent errors are prevalently associated to the complicate geometries, i.e. seawalls and structures with toes and berms. A specific analysis aimed to assess the ANN behavior against such structures is required, in order to evaluate if the wrong predictions are due to either the shortage and/or poor reliability of the available tests or to an intrinsic shortcoming of the model.

For this purpose, it is planned to verify the ANN response when subjected to new out-of-range data, obtained by varying of the toe/berm dimensions from a fixed tested condition.

7.4.3 Contemporary predictions

A specific study about the possibility to contemporary predict all the output parameters was carried out. The unsatisfactory results – which essentially revealed that, at the present state, the ANN performs constantly better when working with only one output neuron – leaded to the impossibility to follow this path.

The main cause was individuated in the shortage of tests reporting the datum relative to more than one output parameter. Besides the enlargement of the database, which of course requires the availability of additional tests, another solution could be the possibility to modify the ANN architecture to include more hidden layers with less neurons. This possibility has not been investigated yet.

8. LIST OF NOTATIONS

ANN(s)	Acronym of Artificial Neural Network(s)
A_c	Armour crest freeboard of the structure
B	Width of the berm
B_h	Width of the horizontally schematised berm
B_t	Toe width
b	Bias, element of the network archtiecture
CF	Complexity factor, describing the degree of complexity of the geometry of a structure in the database
$D_{n,50}$	Nominal rock diameter or typical armour unit size
D	Diameter representative of the armour, when $D_{n,50}$ cannot be defined
G_c	Structure crest width
g	Acceleration due to gravity
$H_{m0,deep}$	Significant incident wave height at deep water
$H_{m0,t}$	Significant incident wave height at the structure toe
h	Water depth at the structure toe
h_b	Berm submergence
h_{deep}	Water depth at deep water
h_t	Water depth above the structure toe
m	Foreshore slope, defined as the cotangent of the angle of the slope itself
mse	Acronym of Mean Square Error
IW	Input Weights, element of the network archtiecture
K_r	wave reflection coefficient
$K_{r,ANN}$	Reflection coefficient predicted by the ANN
$K_{r,\mathrm{s}}$	Experimental reflection coefficient
K_t	wave reflection coefficient
$K_{t,ANN}$	Reflection coefficient predicted by the ANN
$K_{t,\mathrm{s}}$	Experimental reflection coefficient
LCS(s)	Acronym of Low-Crested Structure(s)
$L_{m-1,0,t}$	Wave length based on spectral wave period at the structure toe
IW	Layer Weights, element of the network archtiecture
q	Specific wave overtopping discharge
q_{s}	Experimental overtopping discharge
q_{ANN}	Overtopping discharge predicted by the ANN
Pow	Acronym of Probability of wave overtopping
R^2	Coefficient of determination, see Eq. (4)
R_c	Structure freeborard (negative if the structure is submerged) with the respect to SWL
RF	Reliability factor, depending on the degree of reliability of the test in the database
$rmse$	Acronym of Root Mean Square Error
SWL	Acronym of Still Water Level
s_0	Wave steepness based on spectral wave at the structure toe

$T_{m,deep}$ Mean period from spectral analysis at deep water = m2/m0 (see mn)

$T_{m-1,deep}$ Mean period from spectral analysis at deep water = m-1/m0 (see mn)

$T_{m-1,0,t}$ Spectral wave period at the structure toe

$T_{m,t}$ Mean period from spectral analysis at the structure toe = m2/m0 (see mn)

$T_{p,deep}$ Peak period from spectral analysis at deep water

$T_{p,t}$ Peak period from spectral analysis at the structure toe

VDM Acronym of Van Der Meer

WF Weight factor associated to each test in the database, see Eq. (9)

WI Acronym of Wilmott Index

ZLA Acronym of Zanuttigh and Lykke Andersen

ZVDM Acronym of Zanuttigh and Van Der Meer

α Off-shore slope for structures with a single slope

α_b Berm slope with respect to the horizontal

α_d Slope of the structure downward of the berm

α_{incl} Mean slope of the structure within the run-up and run-down zone, including the berm; the mean angle is defined between ±1.5H_(m0,t)

α_{excl} Mean slope of the structure within the run-up and run-down zone, excluding the berm; the mean angle is defined between ±1.5H_(m0,t)

α_{up} Slope of the structure upward of the berm

β Angle of deviation from the perpendicular wave attack direction

γ_f Roughness factor as found in overtopping research

ξ_0 Breaker parameter based on spectral wave period at the structure toe

9. REFERENCES

Agrawal, J.D. and Deo, M.C., 2004. Wave parameter estimation using neural networks, *Marine Structures*, 17, 536-550.

Ahrens, J. P., 1987. Characteristics of reef breakwaters, Technical Report CERC-87-17, *Coastal Engineering*, Res Ctr., Vicksburg, MS.

Allsop, N. W. H, 1983. Low-crested breakwaters, studies in random waves, *Proc. of Coastal Structures*, Arlington, VA, 94-107.

Allsop, N.W.H. and Channel, A.R., 1989. Wave reflections in harbours: reflection performance of rock armoured slopes in random waves, Report OD vol. 102, *Hydraulic Research*, Wallingford, UK.

Battiti, R., 1992. First and second order methods for learning: Between steepest descent and Newton's method, *Neural Computation*, 4(2), 141-166.

Bierens, R.W.P., 2002. Sedimentation in the Maasmond, dredging prediction by neural networks, *PIANC Bulletin*, No. 109.

Browne, M., Castelle, B., Strauss, D., Tomlinson, R., Blumenstein, M. and Lane, C., 2006. Near-shore swell estimation from a global wind-wave model: Spectral process, linear and artificial neural networks models, *Coastal Engineering*, 54, 445-460.

Bruce, T., Van der Meer, J.W., Franco, L. and Pearson, J., 2006. A comparison of overtopping performance of different rubble mound breakwater armour. *Proc. of XXX International Conference on Coastal Engineering*, San Diego, CA, 4567-4579.

Calabrese, M., Vicinanza, V. and Buccino, M., 2002. Large scale experiments on the behaviour of low crested and submerged breakwaters in presence of broken waves. *Proc. of XXVIII International Conference on Coastal Engineering*, Cardiff, UK, 1900–1912.

Calabrese, M., Di Pace, P. and Buccino, M, 2008. Wave Reflection at Low Crested Breakwaters ranging from submerged to exposed, *Proc. of XXXI International Conference on Coastal Engineering*, Hamburg, D.

Cappietti, L., Clementi, E., Aminti, P. and Lamberti, A., 2006. Piling-up and filtration at low crested breakwaters of different permeability, *Proc. of XXX International Conference on Coastal Engineering*, San Diego, CA, 4957-4969.

Cherkassky, V., Krasnopolsky, V., Solomatine, D.P., and Valdes, J., 2006. Computational intelligence in Earth sciences and environmental applications: issues and challenges, *Neural Networks*, 19, 113-121.

CLASH, 2004. Crest Level Assessment of coastal Structures by full scale monitoring, neural network prediction and Hazard analysis on permissible wave overtopping. EC-contract EVK3-CT-2001-00058. www.clash-eu.org.

Daemen, I. F. R., 1991. Wave transmission at low crested breakwaters. Msc thesis, *Delft Univ. of Technology*, Delft, NL.

Daemrich, K. F. and Kahle, W., 1985. Schutzwirkung von Unterwasserwellenbrechern unter dem Einfluss unregelmässiger Seegangswellen, *Eigenverlag des Franzius-Instituts für Wasserbau und Küsteningenieurwesen*, Heft 64, Hannover, D.

Daemrich, K. F., Mai, S. and Ohle, N., 2001. Wave Transmission at Submerged Breakwaters Waves, *Proc. of the IV International Symposium on Ocean Wave Measurement and Analysis*, San Francisco, CA, 1725-1734.

d'Angremond, K., Van der Meer, J.W., De Jong, R.J., 1996. Wave transmission at low crested structures, *Proc. of XXV International Conference on Coastal Engineering*, Orlando, FL, 3305-3318.

Davidson, M.A., Bird, P.A.D., Bullock, G.N. and Huntley, D.A., 1996. A new non-dimensional number for the analysis of wave reflection from rubble mound breakwaters, *Coastal Engineering*, 28, 93-120.

Dennis, J.E., and Schnabel R.B., 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations, *Englewood Cliffs, NJ: Prentice-Hall.*

Deo, M.C. and Naidu, C.S., 1999. Real time wave forecasting using neural networks. *Ocean Engineering*, 26, 191–203.

Deo, M.C., Jha, A., Chaphekar, A.S. and Ravikant, K., 2001. Neural networks for wave forecasting, *Ocean Engineering*, 28, 889-898.

Deo, M.C., Gondane, D.S. and Kumar, V.S., 2002. Analysis of wave directional spreading using neural networks. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 128, 30-37.

Deo, M.C., Jagdale, S.S., 2003. Prediction of breaking waves with neural networks, *Ocean Engineering*, 30, 1163-1178.

Duda, R. O. and Hart P. E., 1973. Pattern classification and scene analysis, *New York: Wiley*.

EurOtop, 2007. In: Pullen, T., Allsop, N.W.H., Bruce, T., Kortenhaus, A., Schüttrumpf, H., van der Meer, J.W. (Eds.), Wave Overtopping of Sea Defences and Related Structures – Assessment Manual. www.overtopping-manual.com.

Formentin, S.M., Zanuttigh, B., 2012. Le reti neurali artificiali in idraulica marittima. AMS Acta, Id 3210, electronic support. DOI: 10.6092/unibo/amsacta/3210.

Formentin, S.M., Zanuttigh, B. and R. Briganti, 2012. Prediction of the wave reflection coefficient by means of neural network models, *Proc. of XXXIII National Congress of Hydraulics and Hydraulic Construction,* Brescia, electronic format, in Italian, 11 pages.

Formentin, S.M., Zanuttigh, B., 2013. Prediction of wave transmission trough a new artificial neural network developed for wave reflection, *Proc. of VII International Conference on Coastal Dynamics*, Arcachon, FR, electronic support (nr. 69). http://www.coastaldynamics2013.fr/pdf_files/057_Formentin_SaraMizar.pdf.

Garcia, N., Lara, J.L., Losada, I., 2004. 2-D numerical analysis of near-field flow at low-crested permeable breakwaters, *Coastal Engineering* 51(10), 991–1020.

Gironella, X., Sanchez-Arcilla, A., Briganti, R., Sierra, J. and Moreno, L., 2002. Submerged detached breakwaters: towards a functional design. *Proc. of XXVIII International Conference on Coastal Engineering,* Cardiff, UK, 1768–1777.

Goda, Y., Kishira, Y. and Kamiyama, Y., 1975. Laboratory investigation on the overtopping rate of seawalls by irregular waves. *Report of Port and Harbour Research Institute*, 14(4).

Goda, Y., 1985. Random Seas and Design of Maritime Structures, *University of Tokyo Press*, J. ISBN 0-86008-369-1.

González-Escrivá, J.A. and Medina, J.R., 1999. Wave and wind tunnel testing for analysis of run-up and overtopping, *Proc. of International Conference on Coastal Structures*, Santander, SP, 413-420.

Hagan, M.T., and Menhaj M., 1994. Training feed-forward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks*, 5(6), 989-993.

Hagan, M.T., Demuth H.B., and Beale M.H., 1996. Neural Network Design, Boston, MA, *PWS Publishing*.

Herman A., Kaiser R., Niemeyer H. D., 2009. Wind-wave variability in a shallow tidal sea – Spectral modelling combined with neural network methods. *Coastal Engineering* 56(7), doi:10.1016/j.coastaleng.2009.02.007

Hirose, N., Watanuki, A. and Saito, M., 2002. New type units for artificial reef development of ecofriendly artificial reefs and the effectiveness thereof., *Proc. of XXX International Navigation Congress*, Cd ROM. http://it.mathworks.com/products/neural-network/

Johnson, H.K., Karambas, T.V., Avgeris, J., Zanuttigh, B., Gonzalez-Marco, D., and Caceres. I., 2005. Modelling of waves and currents around submerged breakwaters, *Coastal Engineering*, 52, 949–969.

Kalra R., Deo M. C., Kumar R. and Agarwal V. K., 2005. Artificial neural network to translate offshore satellite wave data to coastal locations, *Ocean Engineering*, 32, 1917-1932.

Kramer, M., Zanuttigh, B., van der Meer J. W., Vidal, C. and Gironella, X., 2005. 2D and 3D experiments on low-crested structures, *Coastal Engineering*, 52, 867-88.

Krasnopolsky, V.M., and Chevallier, F., 2003. Some neural network applications in environmental sciences. Part II: Advancing computational efficiency of environmental numerical models, *Neural Networks*, 16, 335-348.

Lara, J.L., Garcia N. and Losada, I.J., 2006. RANS modelling applied to random wave interaction with submerged permeable structures, *Coastal Engineering*, 53, 395-417.

Liriano, S.L. and Daya, R.A., 2001. Prediction of scour depth at culvert outlets using neural networks, *Journal of Hydroinformatics*, 3 (4), 231-238.

Lissev, N., 1993. Influence of the core configuration on the stability of berm breakwaters. Experimental model investigations. Report No. R-6-93, *Department of Structural Engineering, University of Trondheim*, *The Norwegian Institute of Technology*.

Londhe, S. N. and Panchang V., 2006. One-Day Wave Forecasts Based on Artificial Neural Networks, *J. Atmos. Oceanic Technologies*, 23, 1593–1603.
doi: http://dx.doi.org/10.1175/JTECH1932.1

Losada, I.J., Lara J.L., Guanche R. and Gonzales-Ondina J.M., 2008. Numerical analysis of wave overtopping of rubble mound breakwaters, *Coastal Engineering*, 55, 47-62.

Lykke Andersen, T. and Burcharth, H.F., 2004. CLASH Work package 4.4 - D24 Report on additional tests. Part D Berm breakwater tests. On-line available.

Lykke Andersen, T., 2006. Hydraulic Response of Rubble Mound Breakwaters: Scale Effects - Berm Breakwaters. *Aalborg University*, PhD thesis, 429 pp.

Lykke Andersen, T. and Burcharth, H. F., 2009. Three-dimensional investigations of wave overtopping on rubble mound structures, *Coastal Engineering*, 56, 180-189.

Makarynskyy, O. Kuhn, M. Makarynska, D. and Featherstone, W.E., 2004-a. The use of artificial neural networks to retrieve sea-level information from remote data sources, *Jekeli, C, and Bastos, L. and Fernandes, J. (ed), Gravity, Geoid and Space Missions IAG International Symposium*, Porto, P. International Association of Geodesy.

Makarynskyy, O., Makarynska, D., Kuhn, M. and Featherston, W., 2004-b. Predicting sea level variations with artificial neural networks at Hillarys Boat harbor, Western Australia, *Estuarin, Coastal and Shelf Science*, 61(2), 351-360.

Makarynskyy, O., Makarynska, D., Kuhn, M. and Featherston, W., 2005-a. Using artificial neural networks to estimate sea level in continental and island coastal environment. In Cheng, L. and Yeow, K. (ed.), *Hydrodynamics IV: Theory and Applications*, 451-457. London: Taylor & Francis Group.

Makarynskyy, O., Makarynska, D., Rusu, E.and Gavrilov, A., 2005-b. Filling gaps in wave records with artificial neural networks. *In Guedes-Soares, C. (ed.), Maritime Transportation and Exploitation of Ocean and Coastal Resources IMAM*, 26 – 30 September 2005. Lisboa, Portugal: Balkema.

Marquardt D.W., 1963. An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431-441.

Mase, H., Sakamoto, M. and Sakai, T, 1995. Neural network for stability analysis of rubble-mound breakwaters, *Journal of Waterway, Port, Coastal and Ocean Engineering*, 121, 294-299.

Mase, H., Kitano, T., 1999. Prediction model for occurrence of impact wave force, *Ocean Engineering*, 26, 949-961.

Medina, J.R., 1998. Wind effects on run-up and breakwater crest design, *Proc. of XXVI International Conference on Coastal Engineering*, Copenhagen, DK, 1068-1081.

Medina, J.R., 1999. Neural network modelling of runup and overtopping, *Proc. of Coastal Structures*, Santander, SP, 421-429.

Medina, J.R., González-Escrivá, J.A., and Garrido, J., 2001. Zeebrugge model tests performed in the UPV, OPTICREST report, *Universidad Politécnica de Valencia*, SP.

Medina, J.R, González-Escrivá, J.A., Garrido, J. and De Rouck, J., 2002, Overtopping analysis using neural networks, *Proc. of XXVIII International Conference on Coastal Engineering*, Cardiff, UK.

Melito, I., Melby, J.A., 2002. Wave runup, transmission, and reflection for structures armoured with CORE-LOC, *Coastal Engineering*, 45, 33–52.

Medina, J.R, Garrido, J., Gómez-Martín, E. and Vidal, C., 2003. Armour damage analysis using neural networks, *Proc. of the International Conference on Coastal Structures, Portland*, OR, 236-248.

Muttray, M., Oumeraci, H. and Ten Oever, E. 2006. Wave reflection and wave run-up at rubble mound breakwaters. *Proc. of XXX International Conference on Coastal Engineering*, San Diego, CA.

Oumeraci, H., Kortenhaus A. and Haupt R. 2001. Untersuchung zur Abminderung des Wellenüberlaufs bei senkrechten Wänden durch Wellenabweiser, Report Nr. 865, *Technishe Universität Braunschweig*, Leichtweiß-Institut für Wasserbau, Abteilung Hydromechanik und Küsteningenieurwesen.

Oumeraci, H., Kortenhaus A. and Haupt R. 2004. Überarbeitung des Bemessungskonzepts für die Hochwasserschutzwände des privaten Hochwasserschutzes im Hamburger Hafen, Report Nr. 860, *Technishe Universität Braunschweig*, Leichtweiß-Institut für Wasserbau, Abteilung Hydromechanik und Küsteningenieurwesen.

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

Panizzo, A., Briganti, R., van der Meer, J.W. and Franco, L., 2003. Analysis of wave transmission behind low-crested structures using neural networks, *Proceedings of the International Conference on Coastal Structures*, Portland, OR, 555-566.

Panizzo, A. and Briganti, R., 2007. Analysis of wave transmission behind low-crested breakwaters using neural networks, *Coastal Engineering*, 54, 643-656. doi:10.1016/j.coastaleng.2007.01.001

Pearson, J., Bruce, T., Franco, L. And Van der Meer, J.W, 2004. Report on additional tests, part B, CLASH WP4 report, *University of Edinburgh*, Edinburgh, UK.

Powell, K.A. and Allsop N. W. H, 1985. Low-crest breakwaters, hydraulic performance and sta-bility, Report SR57, *Hydraulic Research*, Wallingford, UK.

Pozueta, B., Van Gent, M.R.A., Van den Boogaard, H.F.P., Medina, J.R., 2004. Final report on generic prediction method, CLASH WP8 report, *Delft Hydraulics*, NL.

Rao, S. and Mandal, S., 2005. Hindcasting of storm waves using neural networks, *Ocean Engineering*, 32, 667-684.

Requejo, S., Vidal, C., Losada, I.J., 2002. Modelling of wave loads and hydraulic performance of vertical permeable structures, *Coastal Engineering*, 46, 249-276.

Riedmiller, M., and H. Braun, 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Proceedings of the IEEE International Conference on Neural Networks*, 1993, San Francisco, CA.

Ruol, P., Faedo, A. and Paris, A., 2004. Physical model study of water piling-up behind low crested structures. *Proc. of XXIX International Conference on Coastal Engineering*, Lisbon, P, 4165-4177.

Seabrook S.R. and Hall, K.R., 1998. Wave transmission at submerged rubble mound breakwaters, *Proc. of XXVI International Conference on Coastal Engineering*, Copenhagen, DK, 2000-2013.

Seelig, W. N., 1980. Two-dimensional test of wave transmission and relfection characteristics of laboratory breakwaters. *Technical report No. 80-1, U.S. Army Coastal Engineering Res. Center*, Fort Belvoir, VA.

Smith, G.M., Seijffert, J.W., and van der Meer, J.W., 1994. Erosion and overtopping of a grass dike: large scale model tests, *proceedings of the XXIV International Conference on Coastal Engineering*, Kobe, J, 2639-2652.

Steendam, G.J., Van der Meer, J.W., Verhaeghe, H., Besley, P., Franco, L. and van Gent, M., 2004. The international database on wave overtopping, *Proc. of XXIX International Conference on Coastal Engineering*, Lisbon, P, 4301-4313.

TAW, 2002. Technical report wave run-up and wave overtopping at dikes, *Technical Advisory Committee on Flood Defense*, NL.

Tolman, H. L., Krasnopolsky V. M., and Chalikov, D., 2005. Neural network approximations for nonlinear interactions in wind wave spectra: direct mapping for wind seas in deep water, *Ocean Modelling*, 8, 253-278.

Tsai, C.-P. and Lee, T.L., 1999. Back-propagation neural network in tidal level forecasting, *Journal of Waterway, Port, Coastal and Ocean Engineering*, 125(4), 195-202.

Tsai, C.-P., Hsu, J.R.-C., Pan, K.-L., 2000. Prediction of storm-built beach profile parameters using neural networks, *Proceedings of the XXVII International Conference on Coastal Engineering*, Sydney, AUS, 3048-3061.

Tsai, C.P., Lin, C. and Shen, J.-N, 2002. Neural networks for wave forecasting among multi stations, *Ocean Engineering*, 29, 1683-1695.

Van der Meer, J.W., 1988. Rock slopes and gravel beaches under wave attack. PhD thesis, *Delft University of Technology*, Delft, NL. Also *Delft Hydraulics Publication* no. 396.

Van der Meer, J.W., Wang, B., Wolters, A., Zanuttigh, B. and Kramer, M, 2003. Oblique wave transmission over low-crested structures. *Proc. of Coastal Structures*, Portland, Oregon, 567-579.

Van der Meer, J.W., Briganti, R., Zanuttigh, B. and Wang, B., 2005. Wave transmission and re-flection at low crested structures: design formulae, oblique wave attack and spectral change, *Coastal Engineering*, 52, 915-929.

Van der Meer, J.W., Verhaeghe H. and Steendam, G. J., 2008. The new wave overtopping database for coastal structures, *Coastal Engineering*, 56, 108-120.

Van der Meer, J.W., Verhaeghe, H. and Steendam, G.J. 2009. The new wave overtopping database for coastal structures. *Coastal Engineering* 56, 108–120.

Van der Meer, J.W., Bruce T., Allsop W., Franco L., Kortenhaus A., Pullen T. and Schüttrumpf H. 2013. EurOtop revisited. Part 1: sloping structures. *Proceedings of ICE, Coasts, Marine Structures and Breakwaters*, Edinburgh, UK.

Van Gent, M.R.A. and van den Boogaard, H.F.P., 1998. Neural network modelling of forces on vertical structures, *Proc. of XXVI International Conference on Coastal Engineering*, Copenhagen, DK, 2096-2123.

Van Gent, M.R.A., van den Boogaard, H.F.P., Pozueta B. and Medina, J.R., 2007. Neural network modelling of wave overtopping at coastal structures, *Coastal Engineering*, 54, 586-593. doi:10.1016/j.coastaleng.2006.12.001.

Van Oosten, R.P. and Peixò Marco J. 2005. Wave transmission at various types of low-crested structures using neural networks. MsC Thesis, *TU Delft, Faculty of Civil Engineering and Geosciences, Hydraulic Engineering*.

Verhaeghe, H., 2005. Neural network prediction of wave overtopping at coastal structures, PhD thesis, *Universiteit Gent*, Gent, BE.

Verhaeghe, H., De Rouck, J. and van der Meer, J.W., 2008. Combined classifier–quantifier model: a 2-phases neural model for prediction of wave overtopping at coastal structures, *Coastal Engineering*, 55, 357-374. doi:10.1016/j.coastaleng.2007.12.002

Victor L., 2012. Optimization of the hydrodynamic performance of overtopping wave energy converters: experimental study of optimal geometry and probability distribution of overtopping volumes. PhD dissertation, *Universiteit Gent*, BE.

Wilmott, C.J. 1981. On the validation of models. *Physical Geography* 2, 184-194.

Westra, M.R., Van Vledder, G.Ph., Van Banning, G.K.F.M., Hurdle, D.P., 2002. Predicting water levels using artificial neural networks, *Proc. of XXVIII International Conference on Coastal Engineering*, Cardiff, UK, 1292-1302.

Wurjanto, A. and Kobayashi, N, 1993. Irregular wave reflection and runup on permeable slopes, *Journal of Waterway, Port, Coastal, Ocean Engineering*, 119 (5), 537-557.

Zanuttigh, B. and Lamberti A., 2006. Experimental analysis and numerical simulations of waves and current flows around low-crested coastal defense structures, Journal of Waterway, Port, Coastal and Ocean Engineering, 132 (1), 10-27.

Zanuttigh B. and J.W. Van der Meer, 2008. Wave reflection from coastal structures in design conditions. *Coastal Engineering*, 55, 771-779. doi:10.1016/j.coastaleng.2008.02.009

Zanuttigh, B., Van der Meer, J.W., Lykke Andersen, T., Lara J.L. and Losada, I.J., 2010. Analysis of wave reflection from structures with berms through an extensive database and 2DV numerical modelling, *Proc. of XXXI International Conference on Coastal Engineering*, Hamburg, D, 4, 3285-3297.

Zanuttigh, B. and Lykke Andersen T., 2010. Wave reflection in 3D conditions, Coastal Engineering, 57, 531-538.

Zanuttigh, B., Formentin, S.M. and Briganti R. 2013. A Neural Network for the prediction of wave reflection from coastal and harbor structures. *Coastal Engineering* 80, 49-67.

Zanuttigh B., Formentin S.M. and Van der Meer J.W., 2014. Advances in modelling wave-structure interaction through Artificial Neural Networks, *Proc. of XXXIV International Conference on Coastal Engineering*, Seoul, ROK, in press.

Zelt, J.A. and Skjelbreia, J.E., 1992. Estimating incident and reflected wave field using an arbitrary number of wave gauges, *Proc. of XXIII International Conference on Coastal Engineering*, Venice, I, 1, 777-789.

Zhang, Z., Li, C.-W., Qi, Y., and Li, Y.-S., 2006. Incorporation of artificial neural networks and data assimilation techniques into a third-generation wind-wave model for wave forecasting, *Journal of Hydroinformatics*, 8, 65–76.