



HAL
open science

Using Random forest and Gradient boosting trees to improve wave forecast at a specific location

Aurélien Callens, Denis Morichon, Stéphane Abadie, Matthias Delpy, Benoit Liquet

► **To cite this version:**

Aurélien Callens, Denis Morichon, Stéphane Abadie, Matthias Delpy, Benoit Liquet. Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. *Applied Ocean Research*, 2020, 104, pp.102339. 10.1016/j.apor.2020.102339 . hal-03011157

HAL Id: hal-03011157

<https://univ-pau.hal.science/hal-03011157>

Submitted on 14 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Using Random forest and Gradient boosting trees to improve wave forecast at a specific location

Aurélien Callens^{a,*}, Denis Morichon^b, Stéphane Abadie^b, Matthias Delpey^c, Benoit Liquet^{a,d}

^aUniversité de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LMAP, Pau, France

^bUniversité de Pau et des Pays de l'Adour, E2S UPPA, SIAME, Anglet, France

^cCentre Rivages Pro Tech SUEZ EAU FRANCE, Bidart, France

^dDepartment of Mathematics and Statistics, Macquarie University, Sydney, Australia

Abstract

The main objective is to present alternative algorithms to neural networks when improving sea state forecast by numerical models considering main spectral bulk parameters at a specific location, namely significant wave height, peak wave period and peak wave direction. The two alternatives are random forest and gradient boosting trees. To our knowledge, they have never been used for error prediction method. Therefore, their performances are compared with the performances of the usual choice in the literature: neural networks. We showed that the RMSE of the variables updated with gradient boosting trees and random forest are respectively 20 and 10% lower than the RMSE obtained with neural networks. A secondary objective is to show how to tune the hyperparameter values of machine learning algorithms with Bayesian Optimization. This step is essential when using machine learning algorithms and can improve the results significantly. Indeed, after a fine hyperparameter tuning with Bayesian optimization, gradient boosting trees yielded RMSE values in average 8% to 11% lower for the correction of significant wave height and peak wave period. Lastly, the potential benefits of such corrections in real life application are investigated by computing the extreme wave run-up ($R_{2\%}$) at the study site (Biarritz, France) using the data corrected by the different algorithms. Here again, the corrections made by random forest and gradient boosting trees provide better results than the corrections made by neural networks.

Keywords: Artificial neural networks, Data assimilation, Error prediction, Gradient boosting trees, Random forest, Wave forecasting.

Preprint submitted to Elsevier

November 2019

*Corresponding author

Email address: aurelien.callens@univ-pau.fr (Aurélien Callens)

30 **1. Introduction**

31 Nowadays, numerical wave models are routinely used to forecast wind generated
32 waves. Although they provide satisfactory predictions at a regional scale and during
33 mean wave conditions, it has been shown that they are less accurate for forecasting at
34 a specific location (Londhe et al., 2016) and have a tendency to underestimate wave
35 height during energetic wave conditions. This underestimation has been observed in
36 different state-of-the-art wave models: see the work of Arnoux et al. (2018) for Wind
37 Wave Model II (WWMII) and WAVEWATCH-III (WW3) ; Rakha et al. (2007) for
38 WAVE Model (WAM) and Moeini et al. (2012) for the Simulating WAVes Nearshore
39 model (SWAN). The errors in wave predictions are mainly due to inaccuracies in the
40 wind input that forces the model. The winds used as forcing are numerically simulated
41 and are known to underestimate high wind speeds (Moeini et al., 2012). This results
42 in the underestimation of wave parameters by numerical wave models. Simplifying
43 assumptions, approximations employed in the modeling process, discretization of the
44 domain and a potentially wrong parametrization of the model can also be sources of
45 inaccuracies in wave model predictions (Babovic et al., 2001, 2005).

46 When observation data are available, data assimilation can be used to improve the
47 predictions made by numerical models. There are 4 main categories of data assimila-
48 tion procedures (Refsgaard, 1997; Babovic et al., 2001): updating the input parameters,
49 updating the state variables, updating the model parameters and finally updating the
50 output parameters. The last procedure is called "Error prediction" method and is the
51 most suitable approach to improve model predictions of different output variables at
52 a specific location (Babovic et al., 2005). This procedure presents several advantages
53 comparing to the other data assimilation procedures. First, it covers inaccuracies com-
54 ing from all sources because it improves directly output variables. In addition, it can
55 use a combination of external variables such as meteorological or wind data to increase
56 the accuracy of the predictions. Lastly, it is easy to implement because it consists in
57 only three steps and does not require multiple runs of numerical wave model. First, the
58 deviations between the modeled values and measured values are computed. Then, ma-

chine learning algorithms are used to forecast these deviations. Finally the deviations predicted by the algorithms are incorporated to the predictions of the numerical model for the next time steps, resulting in a more accurate wave forecast.

This method has been successfully applied on hindcast data (Makarynsky et al., 2005; Deshmukh et al., 2016) and has even been implemented in real time setting in the works of Babovic et al. (2001) and Londhe et al. (2016). To our knowledge, only artificial neural networks have been tested to forecast the errors in the data assimilation. However, according to the so-called “No Free Lunch” theorem, there is no single model that works best for all problems (Wolpert, 2002). It is therefore necessary to try multiple models and find the one that works best for our particular problem. The performance of artificial neural networks must be compared with other algorithms in the data assimilation task. Random forest and gradient boosting trees are strong candidates for this comparison. Indeed, these two methods are known for their performance and unlike neural networks, they also provide valuable information by computing the predictive power of each variable used as input. The predictive power or variable importance refers to how much a model relies on that variable to make accurate predictions. A variable with high predictive power means that its values have a significant impact on the prediction values. By contrast, a variable with low predictive power have a limited impact on the prediction values and it can be subtracted from the model to make it simpler and faster.

To explore the performance of random forest and gradient boosting trees, we use as a test case the Basque coast (South west of France). Every winter, the basque coast faces numerous coastal flooding events. To prevent and mitigate the risk of flooding, wave forecast are used to compute the extreme run-up values either by using parametric models such as the formula of Stockdon et al. (2006) or process based models such as Xbeach (Vousdoukas et al., 2012; de Santiago et al., 2017). In both cases, the accuracy of this forecast is of utmost importance as the issuing of the early warning depends on it, especially during energetic wave climate where coastal flooding risk is the highest. In this study, we employ the error prediction method with the different machine learning algorithms and use local meteorological conditions and measured wave parameters from a local buoy to improve the wave forecast. Lastly, we investigate the potential

90 benefits of using such corrections in the computation of extreme run-up values.

91 This study aims to present two alternatives (random forest and gradient boosting
92 trees) to neural networks by comparing their performances when improving regional
93 numerical models. A secondary objective is to show how to tune the hyperparameter
94 values of machine learning algorithms with Bayesian Optimization. In machine learn-
95 ing, a hyperparameter is a parameter whose value is specified by the user before the
96 learning process begins, it will affect how well a model trains and therefore it will have
97 a non negligible impact on the final results. Bayesian optimization is an efficient hy-
98 perparameter optimization algorithm and it is widely used to optimize the results of
99 any given machine learning method.

100 Lastly, we investigate if the error prediction method makes a difference in a real
101 application such as the computation of extreme run-up for the beach of Biarritz. Section
102 2 will introduce the study area, the data and all the statistical methods used. Results will
103 be presented and discussed in Section 3. Finally, Section 4 will cover the conclusion.

104 **2. Data and Methods**

105 *2.1. Study site and Data*

106 The Basque coast is a 150 km long rocky coast facing the Bay of Biscay (Figure
107 1). Every winter, it is battered by numerous storm events. This results in frequent
108 and sometimes intense coastal flooding which can severely damage seafront infrastruc-
109 tures. The city of Biarritz is particularly affected as the buildings and infrastructures
110 are located right behind a sea wall that is located at the top of the beach. The damages
111 associated with coastal flooding are costly for nearshore cities which try to prevent
112 and mitigate the risks by developing early warning systems. Such systems rely on the
113 knowledge of the sea state and its forecast.

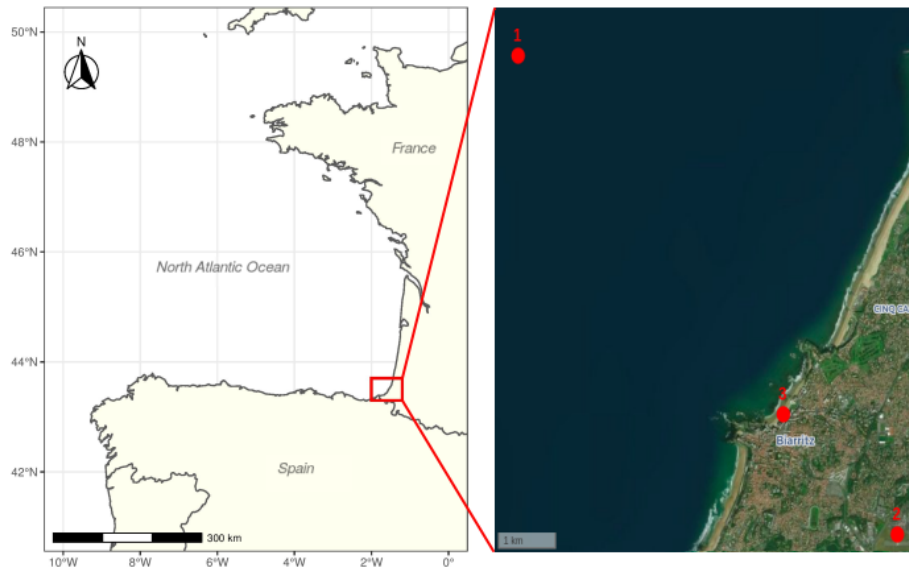


Figure 1: Map showing the location of the study site. The red dots show of the locations of the directional wave buoy (1), the meteorological station (2) and the beach called "Grande Plage de Biarritz" (3).

114 This work focuses on the forecast improvement of three wave integrated parameters
 115 which describe the sea state in this area: the significant wave height (H_S), the peak
 116 period (T_p) and the peak wave direction (θ_p). Direct measurements of these parameters
 117 are obtained from the National Center for Archiving Swell Measurements (L'her et al.,
 118 1999). They were made by a directional wave rider buoy (DWR MKIII) operated by
 119 the Centre for Studies and Expertise on Risks, Environment, Mobility, and Urban and
 120 Country Planning (CEREMA) and the University of Pau and Pays de l'Adour (UPPA).
 121 The buoy is located a few miles off the Basque Coast (Figure 1) at 50 meters water
 122 depth. Since its deployment in 2009, this buoy have been recording the parameters
 123 of interest every 30 minutes. The measuring range of this buoy is [-20m; 20m] for
 124 heave motion, [1.6s; 30s] for wave period and [0°; 360°] for wave direction. It has a
 125 resolution of 1 cm in heave motion and a directional resolution of 1.5°. To be consistent
 126 with the numerical wave data and meteorological data, a 1 hour time step was adopted
 127 for the buoy data.

128 The three parameters simulated at the buoy coordinates by the Meteo-France WAM

129 model were provided by the Copernicus Marine Environment Monitoring Service. This
130 reanalysis ("ibi_reanalysis_wav_005_006") covers the period 2007-2019 with a hourly
131 time-step. The MFWAM model is derived from the third generation wave model WAM
132 (Group, 1988). It is forced by wind fields obtained from a regional numerical weather
133 prediction model (AROME). A more complete description of the MFWAM model can
134 be found in Lefèvre and Aouf (2012).

135 Meteorological data, including average wind speed above 10 meters, wind direction
136 and atmospheric pressure were furnished by the French national meteorological service
137 MétéoFrance. The data were collected hourly by the meteorological station of the
138 Biarritz airport, located only a few kilometers from the study site (Figure 1). It covers
139 the period ranging from 2013-01-01 to 2018-12-31. By assembling the wave buoy
140 data, the wind wave parameters and the meteorological data we obtain a dataset of
141 41439 hourly observations ranging from 2013-01-01 to 2018-12-31.

142 In this work, we are improving the wave forecast by correcting the systematic errors
143 of the wind wave model. Therefore, we are not considering any temporal effects while
144 improving H_S , T_p and θ_p . The dataset was randomly divided into 2 parts: the training
145 part containing 70% of the observations ($n = 28797$) and the testing part containing
146 the remaining 30% ($n = 12342$).

147 2.2. Error prediction method

148 The error prediction method consists in three steps:

- Step 1: Deviations between model predictions and measured values are computed:

$$E_{model} = X_{measured} - X_{modeled},$$

149 where E_{model} is the error of the model, $X_{measured}$ is the measured value of an
150 output variable provided by the wave buoy and $X_{modeled}$ is the value of the same
151 variable computed by the wave model.

- Step 2: E_{model} is predicted with an appropriate supervised machine learning algorithm.

- 154 • Step 3: The predicted error is added to the prediction of the wave model to obtain
155 an updated numerical prediction:

$$X_{updated} = X_{modeled} + E_{predicted},$$

156 where $X_{updated}$ is the updated prediction of wave model and $E_{predicted}$ is the pre-
157 dicted error given by the supervised learning method.

158 This method is repeated separately for each output variable to improve (H_s, T_p, θ_p) .
159 The performance of this data assimilation method relies on two things: the quantity of
160 data and the machine learning algorithm used. Since the machine learning algorithm
161 are generally more suited to interpolate rather than extrapolate, the available data for
162 learning process should cover as much as possible the range of all the probable events
163 in the study area. Concerning the learning method, only neural networks have been
164 used for the step (2) of the error prediction method to our knowledge(Makarynskyy
165 et al., 2005; Moeini et al., 2012; Londhe et al., 2016). Because we want to compare
166 the performance between different machine learning algorithms, we use random forest
167 and gradient boosting trees. All the tested algorithms use the same input variables
168 to improve the model accuracy: the three wave parameters (H_s, T_p, θ_p) given by the
169 numerical model, the atmospheric pressure, the wind direction and speed.

170 2.3. Neural networks

171 Artificial neural networks have been extensively used in the domain of wave mod-
172 elling (Deo et al., 2001; Makarynskyy et al., 2002; Makarynskyy, 2005; Mandal and
173 Prabakaran, 2006) or wave parameters assimilation (Makarynskyy et al., 2005; Moeini
174 et al., 2012; Londhe et al., 2016). It is why technical details will be avoided in this
175 study and only the general concepts will be presented. The readers can find more de-
176 tails and information on the working of neural networks in Liang and Bose (1996) or
177 Friedman et al. (2001).

178 The most common class of neural networks is the multilayer perceptron. The neu-
179 rons in this network are organized in three layers: the input layer that receive the input
180 variables, the output layer that performs the final predictions and between these two

181 layers there is the hidden layer. Neurons in the hidden layer transmit the signal to the
182 output layer by transforming the weighted sum of the neurons present in the input layer
183 with a non linear function called activation function. The weights between each neuron
184 of the network are adjusted through the iterative process of backpropagation to mini-
185 mize the error between the variable we want to predict and the variable predicted by
186 the network (output layer).

187 As other machine learning methods, hyperparameters need to be specified before
188 the training of neural networks. Some hyperparameters control the network architec-
189 ture (number of neurons, layers, activation function used, etc...) while others control
190 the training process (learning rate, batch size, number of epochs, etc...). Hyperparam-
191 eters must be tuned carefully in order to achieve optimal results with neural networks.

192 2.4. *Tree based algorithms*

193 Unlike neural networks, random forest and gradient boosting have never been used
194 in the error prediction method. They are state-of-the-art ensemble learning techniques
195 for classification and regression tasks. An ensemble learning technique commonly
196 refers to a method that combines the predictions from multiple machine learning algo-
197 rithms, called base learners, to produce more accurate predictions.

198 Random forest is an algorithm that builds many decision trees in parallel. These
199 trees are the base learners for random forest and they have the following characteristics:

- 200 • Each tree is built using a different bootstrap sample of the data-set. This mecha-
201 nism is called bagging.
- 202 • At each node, a given number (hereafter "mtry") of variables are randomly sam-
203 pled as candidates at each split. The best split point is then selected within this
204 random set of variables. This process is called feature sampling. The value
205 "mtry" is fixed before growing the forest.
- 206 • Unlike the classification and regression trees of Breiman et al. (1984), the trees
207 in random forest are fully grown (no pruning step).

208 Bagging and feature sampling are the core principles of random forest. They are
209 two randomizing mechanisms which ensure that the trees are independent and are less

210 correlated with each other. The final prediction of a random forest is obtained by
211 averaging the results of all the independent trees in case of regression or using the
212 majority rule in case of classification.

213 The most important hyperparameters in random forest are the number of trees and
214 "mtry": the number of variables randomly sampled as candidates at each split when
215 building the trees.

216 Gradient boosting is an algorithm that trains many weak learners sequentially to
217 provide a more accurate estimate of the response variable. A weak learner is a machine
218 learning model that perform slightly better than chance. In case of gradient boosting
219 trees, the weak learners are shallow decision trees. Each new tree added to the ensemble
220 model (combination of all the previous trees) minimizes the loss function associated
221 with the ensemble model. The loss function depends on the type of the task performed
222 and can be chosen by the user. For regression, the standard choice is the squared loss.
223 By adding sequentially trees that minimize the loss function (i.e. follow the gradient
224 of the overall loss function), the overall prediction error decreases. Technical details
225 about gradient boosting trees can be found in (Friedman, 2001).

226 Many hyperparameters have to be tuned for gradient boosting trees, some of them
227 control the gradient boosting process, such as the learning rate, the number of trees to
228 be used whereas others regulate the construction process of the trees: minimal node
229 size, sample of the dataset to be used, maximum depth.

230 2.5. Hyperparameter tuning

231 Hyperparameters influence significantly the training of the machine learning algo-
232 rithms and therefore the quality of their predictions. The objective of hyperparameter
233 tuning is to find the values of hyperparameters that yield the lowest error (RMSE in
234 our case) for unseen data. Two types of methods exist to find the optimal values of
235 hyperparameters: uninformed or informed.

236 In uninformed methods, many combinations of hyperparameter values are tested
237 one after the other and the best combination is the one that yields the lowest error
238 on unseen data. The values of hyperparameters are either sampled randomly (random
239 search) or sampled along a grid (grid search). In both cases, each combination tested

240 are independent from another. With grid and random search, it is not guaranteed to find
241 the optimal set of hyperparameters and it usually requires a lot of iterations (combina-
242 tions tested).

243 In informed methods, the results obtained by the past combinations are used to
244 choose the next combination to evaluate. Bayesian optimization algorithm is an in-
245 formed method that aims to minimize an objective function, in our case the errors
246 of the machine learning algorithms on unseen data. First, it builds a probability model
247 (Gaussian process) of the objective function. Then it uses this surrogate model to select
248 the most promising values of hyperparameters to evaluate. Once the promising com-
249 bination of values have been evaluated, the probability model is updated and searched
250 again for the most promising combination. This process is repeated several times. This
251 method is employed in this article because it is very efficient for tuning hyperparameter
252 values and it usually requires less iterations than uninformed methods (Bergstra et al.,
253 2011). In-depth details of this method are given in the works of Snoek et al. (2012);
254 Marchant and Ramos (2012) and Shahriari et al. (2015).

255 2.6. *Training the algorithms*

256 The machine learning algorithms described above are trained to predict the devia-
257 tions of H_s , T_p or θ_p (one model for each variable), using 6 input variables: the three
258 wave parameters (H_s , T_p , θ_p) given by the numerical model, the atmospheric pressure,
259 the wind direction and speed.

260 The neural networks are built and trained with the R package **keras**. The input vari-
261 ables are centered and scaled to improve the result of neural networks and the weights
262 are updated with the adam optimization algorithm (Kingma and Ba, 2014). Random
263 forest and gradient boosting model are fitted in R using respectively the **ranger** pack-
264 age which provide fast implementation of Random Forests (suited for high dimensional
265 data) and the **xgboost** package which is an efficient R implementation of the gradient
266 boosting framework from Chen and Guestrin (2016). The input variables are not cen-
267 tered or scaled before the training of random forest and gradient forest because it does
268 not influence the training of these algorithms.

269 The training is done twice: once with the default values of the hyperparameters in

270 the R packages and once with the optimal values found with the Bayesian optimization
271 method.

272 The best hyperparameter values are found by the means of Bayesian optimization
273 method coupled with a 5-fold cross validation in the training dataset. That is, the
274 training data are split into five equal-sized partitions and a machine learning model is
275 recursively built on four partitions (80% of the training data) with a given hyperpa-
276 rameter combination. A performance metric, in our case the root mean square error is
277 assessed on the remaining partition (20% of the training data). The resulting five per-
278 formance metrics are averaged to provide an estimated out-of-sample performance of
279 the respective hyperparameter combination. The objective function to minimize for the
280 Bayesian optimization method is the average out-of-sample performance value. The
281 Bayesian optimization for our data is performed using the R package **RBayesianOpti-**
282 **mization**. First, random combinations of hyperparameter values are evaluated to serve
283 as search base for the informed method (5 in this study), then an acquisition function
284 (upper confidence bound) is used to find the next combination values to evaluate (this
285 step is repeated 25 times).

286 3. Results and Discussion

287 3.1. Model comparison

288 To assess the accuracy of the numerical model and the proposed corrections, sev-
289 eral metrics are computed including the root mean square error (RMSE), the correlation
290 coefficient, the bias and the scatter index (SI). The bias represents the average error be-
291 tween the observed and modeled data and allows one to detect under or over estimation
292 of the value of one parameter. The scatter index is a measure of the error normalized
293 by the observation values. It is a standard metric for wave model inter-comparison
294 (Londhe et al., 2016). More details about the computation of these two metrics can be
295 found in the work of Mentaschi et al. (2013). The metrics are computed twice: once
296 with the whole test data and once with a subset of the test data where $H_s > 3m$ because
297 the underestimation of H_s is known to become larger above this height (Arnoux et al.,
298 2018).

Table 1: Statistical metrics for the three variables of interest before the hyperparameter tuning. "Ann" stands for artificial neural networks, "Rf" for random forest and "Gb" for gradient boosting tree.

	Hs				Tp				θ_p			
	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.
<i>Computed with all data</i>												
Biais	-0.201	0.005	-0.002	-0.004	0.712	0.002	-0.026	0.005	-0.249	0.698	0.976	0.765
RMSE	0.399	0.306	0.248	0.267	1.839	1.603	1.282	1.388	13.803	12.391	9.749	10.645
SI	0.166	0.148	0.120	0.129	0.153	0.145	0.116	0.125	0.045	0.041	0.032	0.035
Cor	0.954	0.962	0.975	0.972	0.78	0.804	0.880	0.857	0.330	0.366	0.455	0.386
<i>Computed with data where $H_s > 3m$</i>												
Biais	-0.536	-0.156	-0.124	-0.126	0.683	-0.026	-0.090	-0.041	1.925	-0.561	0.052	0.046
RMSE	0.766	0.515	0.420	0.433	1.348	1.083	0.956	1.022	7.351	5.769	4.449	4.931
SI	0.133	0.120	0.098	0.101	0.089	0.083	0.073	0.078	0.024	0.019	0.015	0.016
Cor	0.818	0.857	0.902	0.898	0.832	0.850	0.886	0.869	0.589	0.604	0.790	0.726

299 Table 1 presents the metrics obtained with no assimilation (numerical model) and
300 after a preliminary data assimilation with the three machine learning algorithms. The
301 term "preliminary" refers to the lack of hyperparameter tuning. The learning has been
302 done using the default hyperparameter values given in table 2.

303 For significant wave height, the numerical model shows a negative bias. This indi-
304 cates that the MFWAM model has a tendency to underestimate H_s such as other wind
305 wave models (Moeini et al., 2012; Arnoux et al., 2018). The negative bias increases as
306 the value of H_s becomes larger ($H_s > 3m$), meaning that H_s is more likely to be under-
307 estimated during energetic events. For the peak period, the numerical model shows a
308 positive bias. When $H_s > 3m$, the bias and the RMSE for this parameter are smaller.
309 The predictions of T_p are therefore better during energetic conditions. For wave di-
310 rection, a small bias is observed in average and is greater when the waves are larger.
311 The large difference in RMSE computed with data where $H_s > 3m$ and with all data is
312 explained by the distribution of the wave direction according to the wave height. When
313 the significant wave height is below 2 meters, the wave direction at the buoy is more
314 variable (Figure S1, supplementary material) and the spectral wave model has more
315 difficulties to predict correctly the direction. This can be confirmed by looking at the
316 θ_p errors of the numerical model: we see that they are larger and occur more often

317 when $H_s < 2m$ (Figure S2, supplementary material). A potential explanation of this
318 phenomenon could be that below 2 meters, the sea state is more likely to be influenced
319 by local wind conditions which are difficult to reproduce by the spectral wave model
320 (Rascle and Ardhuin, 2013). When the significant wave height is above 2 meters, the
321 wave directions are a lot less variable and the predictions of the spectral wave model
322 are more accurate.

323 When we look at the metrics computed with all data, we see that the correction
324 made by the three machine learning algorithms removes the bias and greatly reduces the
325 RMSE and the scatter index for H_s and T_p . For θ_p , the mean bias is slightly larger after
326 data assimilation for all algorithms. The correction of the machine learning algorithm
327 could be less efficient for θ_p due to the high variability of the observed deviations
328 they try to model (see the explanation in the paragraph above). However, lower value
329 of RMSE and scatter index and larger correlation coefficients still indicate that the
330 corrected data are closer to the observed values at the buoy.

331 For the metrics computed with data where $H_s > 3m$, the correction does not remove
332 the bias for H_s and T_p but reduces it greatly. For the wave direction, the updated
333 parameters are closer to the reality. Indeed, bias and RMSE obtained by the corrections
334 are smaller than the numerical model and the correlation coefficients are larger for
335 corrected data.

336 For this preliminary assimilation, random forest yields the best results for all the
337 parameters. It reduces the RMSE values computed with all test data by 37.7%, 30%
338 and 29% respectively for H_s , T_p and θ_p . Gradient boosting trees is close second and
339 decreases the RMSE values by 33%, 24.5% and 22.8%. Finally, data assimilation with
340 neural networks decreases the RMSE of H_s , T_p and θ_p by 23%, 12.8% and 10.2%.

341 As stated earlier, the performance of machine learning algorithms might depend
342 on the choice of the hyperparameter values. The Bayesian optimization was therefore
343 performed and optimal values were selected (Table 2). The selected hyperparameter
344 values are quite different from the default values. Indeed, for neural networks, the best
345 results were obtained with more epochs and more neurons in the hidden layer. For
346 random forest, only the number of trees seems to have some effect on the results and
347 models with a large number of trees performs better. Finally, for gradient boosting

Table 2: Default values, ranges and selected value of hyperparameters for the machine learning algorithms

Machine learning algorithms	Hyperparameters	Default value	Range searched	Selected value for H_s	Selected value for T_p	Selected value for Dir
<i>Neural networks</i>	No. of units in hidden layer	$13 (2 \times h + 1)$	{1-40}	26	20	40
	Activation function	sigmoid	{relu, sigmoid, tanh}	sigmoid	sigmoid	relu
	Learning rate	0.001	{0.0001-0.1}	0.021	0.016	0.005
	Epochs	30	{10,30,50,100,150}	50	100	150
	Batch size	32	{16,32,64,128}	32	64	64
<i>Gradient Boosting trees</i>	Number of trees	100	{100-2000}	560	1150	1990
<i>Random forest</i>	Learning rate	0.3	{0.0001-0.3}	0.072	0.028	0.069
	Max depth	6	{1-20}	14	20	20
	Minimal node size	1	{1-15}	7	1	1
	Subsample	1	{0.5-1}	0.57	0.82	0.79
	Col sample	1	{0.5-1}	0.99	0.85	0.9
<i>Random forest</i>	Number of trees	500	{100,200,500,800,1000}	1000	1000	1000
	Mtry	$2 (\sqrt{h})$	{2-6}	2	2	2

Note: h corresponds to the number of input variables (6 in our case).

348 trees, models with a large number of trees and a small learning rate are preferred.

349 Metrics calculated with data corrected by the tuned machine learning algorithms
350 are presented in Table 3. Overall, tuning the hyperparameter values has improved the
351 results of all the algorithms. However, the degree of improvement differs depending
352 on the algorithm. We observe the smallest improvements for random forest where the
353 RMSE of every parameters seems to decrease by less than 1% in average. For neural
354 networks, tuning hyperparameter values has a more significant effect by reducing the
355 RMSE by 2 to 3% in average. The largest effect of tuning the hyperparameters are ob-
356 served with gradient boosting trees. The RMSE is 8 to 11% lower for every parameter.
357 The only exception is θ_p computed with all data where we have a small increase (2%)
358 of RMSE. In general, the mean bias for H_s , T_p and θ_p remains the same before and
359 after hyperparameter tuning expect for the bias of H_s computed when $H_s > 3m$ which
360 is significantly lower after the tuning.

361 For this dataset, gradient boosting algorithm shows the best performances for all
362 parameters. Assimilation with this algorithm decreases the RMSE values computed
363 with all test data by 39.8% for H_s , 33% for T_p and 31% for θ_p . For H_s and θ_p , the
364 reduction are even lower for the RMSE values computed with $H_s > 3m$: 47% for

Table 3: Statistical metrics for the three variables of interest after the hyperparameter tuning. "Ann" stands for artificial neural networks, "Rf" for random forest and "Gb" for gradient boosting tree.

	Hs				Tp				θ_p			
	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.	Numerical model	Ann Corr.	Rf Corr.	Gb Corr.
<i>Computed with all data</i>												
Biais	-0.201	0.026	-0.002	-0.001	0.712	0.007	-0.022	0.003	-0.249	0.790	0.979	0.714
RMSE	0.399	0.300	0.246	0.240	1.839	1.553	1.269	1.231	13.803	12.07	9.646	9.501
SI	0.166	0.144	0.118	0.116	0.153	0.140	0.114	0.111	0.045	0.04	0.032	0.031
Cor	0.954	0.964	0.976	0.977	0.78	.817	0.882	0.889	0.330	0.36	0.461	0.421
<i>Computed with data where $H_s > 3m$</i>												
Biais	-0.536	-0.117	-0.120	-0.099	0.683	0.032	-0.084	-0.051	1.925	-1.114	0.062	0.056
RMSE	0.766	0.495	0.417	0.404	1.348	1.064	0.950	0.943	7.351	5.820	4.412	4.365
SI	0.133	0.117	0.097	0.095	0.089	0.081	0.072	0.072	0.024	0.019	0.015	0.015
Cor	0.818	0.861	0.903	0.908	0.832	0.856	0.888	0.889	0.589	0.609	0.793	0.793

365 the significant wave height and 40% for wave direction. The performances of random
366 forest for H_s , T_p and θ_p are slightly better than the results obtained before tuning the
367 hyperparameters: respectively 38.3%, 30.9%, 30.1%. The performances are also bet-
368 ter for neural networks after hyperparameter tuning: it decreases the RMSE values by
369 24.8% for H_s , 15.5% for T_p and 12.5% for θ_p . The differences in efficiency between
370 neural networks and ensemble learning techniques could be explained by the architec-
371 ture chosen for the neural networks. Indeed, this work shows the results for multilayer
372 perceptrons with only one hidden layer which is the typical choice in the literature
373 (Londhe et al., 2016; Moeini et al., 2012). By choosing an architecture with more hid-
374 den layers, the networks might be able to model more complex phenomena and bring
375 a better improvement for the three wave parameters.

376 The distribution of the errors after the different corrections are presented in the
377 figure 2. For all wave parameters, the distributions of the errors after a correction
378 have narrowed and are now more centered in zero. The differences in performance be-
379 tween algorithms are confirmed with these violin plots. Indeed, when the correction is
380 made with random forest or gradient boosting trees, the distributions of the errors are
381 more narrow than the distributions of the errors obtained with neural networks. The
382 difference in efficiency between random forest and gradient boosting trees is not dis-

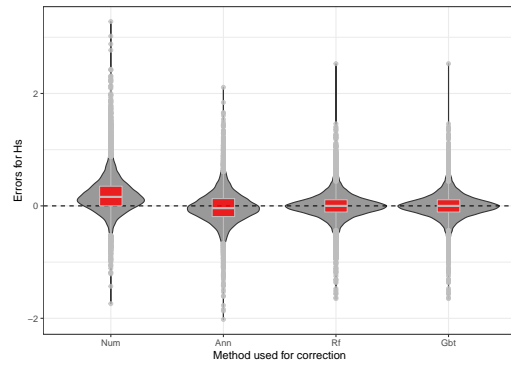
383 tinguishable graphically. It is expected as the metrics of the two algorithms only differ
384 by a few percents. For H_s and T_p , the corrections have also removed the bias observed
385 for numerical model. The large errors of θ_p for the numerical model (Figure ??) are
386 observed when $H_s < 2m$ and are not corrected by the machine learning algorithms.
387 Figures showing the observed values versus the corrected values are available in the
388 supplementary material for the three wave parameters.

389 *3.2. Predictive power of the input variables*

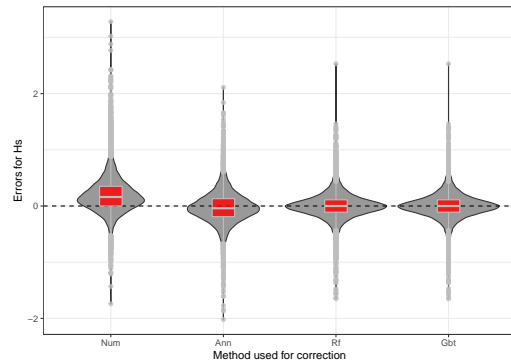
390 In addition to their performance, random forest and gradient boosting algorithms
391 can provide a measure of importance for each variable used as input. This importance
392 indicates the predictive power of the variable. It can be used to sort variable from most
393 to least predictive, allowing one to have more insight on the problem and to perform
394 feature selection when there are too many input variables. The figure 3 shows the
395 importance measure of each variable computed by the random forest depending on the
396 parameter to improve. For H_s and T_p , the most important variables are the value of
397 H_s and T_p modeled by the wind wave model. It is different for the direction where
398 the most important variables are the value of θ_p and H_s given by the model. The
399 predictive power of local meteorological variables is quite low, suggesting that local
400 and instantaneous meteorological variables does not bring valuable information in the
401 assimilation process. The wind wave formation process is not instantaneous and occurs
402 in large regional scale, therefore using meteorological variables from the past (several
403 days before) and from different locations (located in the ocean) could lead to a better
404 predictive power which means better updated wave predictions.

405 *3.3. Example of application*

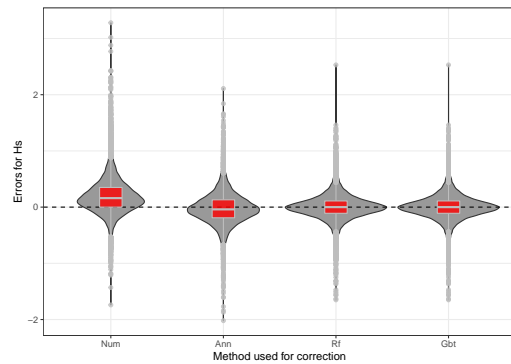
406 To investigate the potential effect of the different corrections in a real case scenario,
407 the extreme wave run-up $R_{2\%}$ at the Grande Plage de Biarritz has been computed for the
408 test period with the Stockdon formula (Stockdon et al., 2006) which uses H_s and T_p and
409 the beach slope as parameters. The beach slope is fixed to 8% according to the work of
410 Morichon et al. (2018). Using the extreme wave run-up calculated with the buoy data
411 as reference, the metrics presented previously have been computed for the numerical



(a) H_s correction



(b) T_p correction



(c) θ_p correction

Figure 2: Distribution of the errors computed between values observed at the buoy and values corrected or not with the different machine learning algorithms. "Num" stands for numerical model (no correction), "Ann" for artificial neural networks, "RF" for random forest and "Gb" for gradient boosting trees. The horizontal lines in the red boxplots represent from top to bottom: the third quartile, the median and the first quartile.

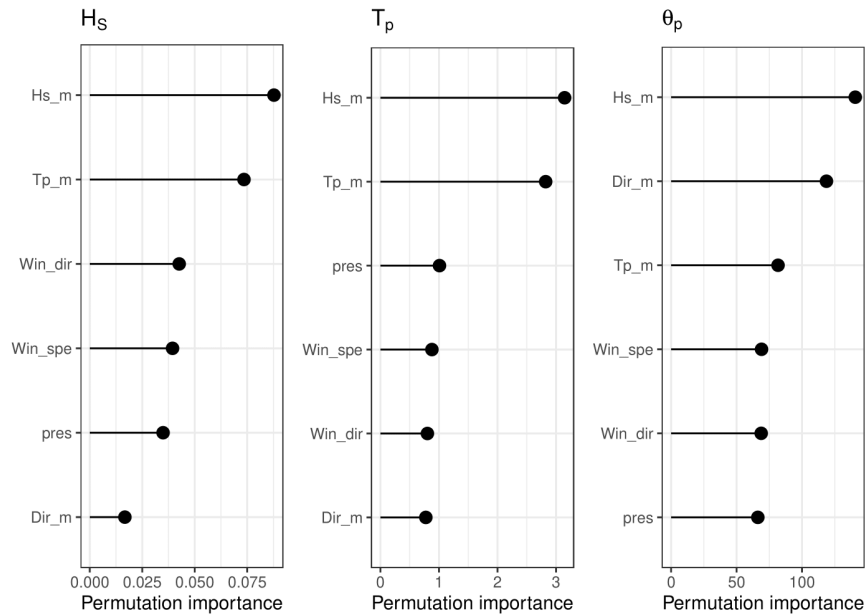


Figure 3: Variable importance for the correction of the three wave integrated parameters.

412 model and the different corrections (Table 4). From this table, it is evident that the data
 413 corrected with machine learning algorithms provide wave run up values that are closer
 414 to the "real" values with lower RMSE, Scatter index and greater correlation coefficient.
 415 Although the bias remains, the correction made by the gradient boosting tree algorithm
 416 decreases the RMSE of the extreme wave run-up by 22% (for all data and data where
 417 $H_s > 3m$). Random forest shows almost the same reduction of RMSE values: 21.5%
 418 for all data and 20.7% for data where $H_s > 3m$. The correction obtained by neural
 419 networks is less efficient: it reduces the RMSE computed with all data and data where
 420 $H_s > 3m$ by 6.2 and 9.9% respectively.

421 4. Conclusion

422 In this work, random forest and gradient boosting trees were employed for the first
 423 time in the error prediction method. These ensemble learning techniques based on
 424 decision trees performed better than neural networks for improving the wave forecast
 425 of the Basque Coast. The correction made by gradient boosting trees yielded the best

Table 4: Statistical metrics of the $R2\%$ calculated with Stockdon's formula. These results are obtained by taking the $R2\%$ computed with buoy data as reference. "Ann" stands for artificial neural networks, "Rf" for random forest and "Gb" for gradient boosting tree.

	Numerical model	Ann Corrected	Rf Corrected	Gb Corrected
<i>Computed with all data</i>				
Biais	0.003	0.019	0.002	0.003
RMSE	0.223	0.209	0.175	0.172
SI	0.145	0.136	0.114	0.112
Cor	0.943	0.950	0.965	0.966
<i>Computed with data where $H_s > 3m$</i>				
Biais	-0.042	-0.030	-0.054	-0.042
RMSE	0.313	0.282	0.248	0.242
SI	0.119	0.108	0.093	0.092
Cor	0.854	0.862	0.897	0.901

426 results for all the wave parameters: it reduced the RMSE values by nearly 40% for
427 H_s , 33% for T_p and 31% for θ_p . The reduction of RMSE values for random forest
428 was only a few percents lower than gradient boosting trees. The corrections made by
429 neural networks were significant but yielded reductions in RMSE not as high as the
430 two ensemble learning techniques: 24.8% for H_s , 15.5% for T_p and 12.5% for θ_p .

431 As expected, tuning the hyperparameters of the machine learning algorithms had a
432 positive effect on the final results. However, the effect of the tuning differed depending
433 on the algorithms. Indeed, random forest was less affected as it only reduced the RMSE
434 values by 1% in average. The tuning had more effect on neural networks reducing the
435 RMSE values by 2 to 3%. Gradient boosting tree algorithm was the most affected by
436 hyperparameter tuning as the results were improved by 8 to 11% in average. One of
437 the main advantage of random forest over gradient boosting trees is that it doesn't need
438 this tuning step in order to yield great results. This is not negligible as hyperparameter
439 tuning step can be time consuming and computationally demanding depending on the
440 complexity of the search (number of hyperparameters).

441 Contrary to neural networks, Random forest and Gradient boosting trees provided
442 valuable insights by giving the predictive power of each input variable. The predictive
443 power of variable brings interpretability to the model and can give a better understand-
444 ing of the variable we try to predict. For example, we know that the significant wave
445 height modelled by the numerical wave model was the most important variable in the
446 correction of the three parameters. In cases where there are a lot of input variables,
447 knowing their associated predictive power helps developing more parsimonious mod-
448 els by keeping the pertinent variables and subtracting the less informative ones from
449 the model.

450 The error prediction method has proven to be useful in improving wave forecast.
451 This had an impact in a real life application by improving the accuracy of the extreme
452 run-up computed at the Grande Plage de Biarritz. Here again the corrections brought by
453 random forest and gradient boosting tree were better than the correction made by neural
454 networks. The decrease in RMSE values was around 22% for the two ensemble tech-
455 niques and 6.2% for the neural networks. Even though the differences in performance
456 might not appear significant, it can make a difference when using these corrections in

457 an early warning system. It is especially true when dealing with storm events where H_s
458 and T_p are large.

459 The differences between machine learning algorithms observed in this article are
460 specific to Biarritz site. The results might differ for another study site. Therefore, we
461 can only advise to test and compare several machine learning algorithms to find the
462 optimal one associated with the site of interest.

463 Finally, the assimilation made in this study did not account for the temporal aspect
464 in the errors of the numerical model, it only corrected systematic errors of the wave
465 model. In the future, this work could be extended by adding input variables containing
466 temporal aspect. This could be the values of a modeled parameter at previous time steps
467 such as the work of Londhe et al. (2016). In this framework, neural networks could
468 perform better as they are known to handle efficiently time series. Other input variables
469 could be also used to improve the wave forecast such as the meteorological data from
470 the past or at different locations. Because the success of the error prediction method
471 depends on the quantity of data, it would be also interesting to perform a sensitivity
472 analysis on the quantity of data used in the training process. This could give us some
473 insights on the minimal quantity of data required to obtain a desirable assimilation
474 procedure.

475 **Acknowledgments**

476 Funding was provided by the Energy Environment Solutions (E2S-UPPA) consor-
477 tium and the BIGCEES project from E2S-UPPA ("Big model and Big data in Compu-
478 tational Ecology and Environmental Sciences"). The authors would like to thank the
479 French national meteorological service "MeteoFrance" and Copernicus Marine Envi-
480 ronment Monitoring Service for providing data.

481 **Reproducibility**

482 Meteorological data used in this article are private and can not be provided by
483 the authors. However, the R code to perform the analysis and an example of data

484 assimilation on wave forecast data (used in operational) are provided in this [Github](#)
485 [repository](#).

486 **References**

487 Arnoux, F., Abadie, S., Bertin, X., Kojadinovic, I., 2018. A database to study storm
488 impact statistics along the Basque Coast. *Journal of Coastal Research* 85, 806–810.

489 Babovic, V., Cañizares, R., Jensen, H.R., Klinting, A., 2001. Neural networks as rou-
490 tine for error updating of numerical models. *Journal of Hydraulic Engineering* 127,
491 181–193.

492 Babovic, V., Sannasiraj, S.A., Chan, E.S., 2005. Error correction of a predictive ocean
493 wave model using local model approximation. *Journal of Marine Systems* 53, 1–17.

494 Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-
495 parameter optimization , 2546–2554.

496 Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J., 1984. *Classification and regres-*
497 *sion trees* Chapman & Hall. New York .

498 Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: *Proceed-*
499 *ings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery*
500 *and Data Mining*, ACM. pp. 785–794.

501 de Santiago, I., Morichon, D., Abadie, S., Reniers, A.J., Liria, P., 2017. A comparative
502 study of models to predict storm impact on beaches. *Natural Hazards* 87, 843–865.

503 Deo, M.C., Jha, A., Chaphekar, A.S., Ravikant, K., 2001. Neural networks for wave
504 forecasting. *Ocean engineering* 28, 889–898.

505 Deshmukh, A.N., Deo, M.C., Bhaskaran, P.K., Nair, T.B., Sandhya, K.G., 2016.
506 Neural-network-based data assimilation to improve numerical ocean wave forecast.
507 *IEEE Journal of Oceanic Engineering* 41, 944–953.

508 Friedman, J., Hastie, T., Tibshirani, R., 2001. *The Elements of Statistical Learning*.
509 volume 1. Springer series in statistics New York.

- 510 Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine.
511 *Annals of statistics* , 1189–1232.
- 512 Group, T.W., 1988. The WAM model—A third generation ocean wave prediction
513 model. *Journal of Physical Oceanography* 18, 1775–1810.
- 514 Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv
515 preprint arXiv:1412.6980 .
- 516 Lefèvre, J.M., Aouf, L., 2012. Latest developments in wave data assimilation, in:
517 *ECMWF Workshop on Ocean Waves*, pp. 25–27.
- 518 L’her, J., Goasguen, G., Rogard, M., 1999. CANDHIS database of in situ sea states
519 measurements on the French coastal zone, in: *The Ninth International Offshore and*
520 *Polar Engineering Conference*, International Society of Offshore and Polar Engi-
521 neers.
- 522 Liang, P., Bose, N.K., 1996. *Neural network fundamentals with graphs, algorithms and*
523 *applications*. Mac Graw-Hill .
- 524 Londhe, S.N., Shah, S., Dixit, P.R., Nair, T.M.B., Sirisha, P., Jain, R., 2016. A Cou-
525 pled Numerical and Artificial Neural Network Model for Improving Location Spe-
526 cific Wave Forecast. *Applied Ocean Research* 59, 483–491. doi:10.1016/j.apor.
527 2016.07.004.
- 528 Makarynsky, O., 2005. Neural pattern recognition and prediction for wind wave data
529 assimilation. *Pac Oceanogr* 3, 76–85.
- 530 Makarynsky, O., Pires-Silva, A.A., Makarynska, D., Ventura-Soares, C., 2002. Arti-
531 ficial neural networks in the forecasting of wave parameters, in: *7th International*
532 *Workshop on Wave Hindcasting and Forecasting*. Banff, Alberta, Canada, pp. 514–
533 522.
- 534 Makarynsky, O., Pires-Silva, A.A., Makarynska, D., Ventura-Soares, C., 2005. Arti-
535 ficial neural networks in wave predictions at the west coast of Portugal. *Computers*
536 *& Geosciences* 31, 415–424.

- 537 Mandal, S., Prabakaran, N., 2006. Ocean wave forecasting using recurrent neural
538 networks. *Ocean engineering* 33, 1401–1410.
- 539 Marchant, R., Ramos, F., 2012. Bayesian optimisation for intelligent environmental
540 monitoring, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and
541 Systems, IEEE. pp. 2242–2249.
- 542 Mentaschi, L., Besio, G., Cassola, F., Mazzino, A., 2013. Problems in RMSE-based
543 wave model validations. *Ocean Modelling* 72, 53–58.
- 544 Moeini, M.H., Etemad-Shahidi, A., Chegini, V., Rahmani, I., 2012. Wave data assim-
545 ilation using a hybrid approach in the Persian Gulf. *Ocean Dynamics* 62, 785–797.
- 546 Morichon, D., de Santiago, I., Delpey, M., Somdecoste, T., Callens, A., Liquet, B.,
547 Liria, P., Arnould, P., 2018. Assessment of flooding hazards at an engineered beach
548 during extreme events: Biarritz, SW France. *Journal of Coastal Research* 85, 801–
549 805.
- 550 Rakha, K.A., Al-Salem, K., Neelamani, S., 2007. Hydrodynamic atlas for Kuwaiti
551 territorial waters. *Kuwait Journal of Science and Engineering* 34, 143.
- 552 Rasclé, N., Ardhuin, F., 2013. A global wave parameter database for geophysical
553 applications. part 2: Model validation with improved source term parameterization.
554 *Ocean Modelling* 70, 174–188.
- 555 Refsgaard, J.C., 1997. Validation and intercomparison of different updating procedures
556 for real-time forecasting. *Hydrology Research* 28, 65–84.
- 557 Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2015. Taking the
558 human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*
559 104, 148–175.
- 560 Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of ma-
561 chine learning algorithms, in: *Advances in Neural Information Processing Systems*,
562 pp. 2951–2959.

- 563 Stockdon, H.F., Holman, R.A., Howd, P.A., Sallenger Jr, A.H., 2006. Empirical pa-
564 rameterization of setup, swash, and runup. *Coastal engineering* 53, 573–588.
- 565 Vousdoukas, M.I., Ferreira, Ó., Almeida, L.P., Pacheco, A., 2012. Toward reliable
566 storm-hazard forecasts: XBeach calibration and its potential application in an oper-
567 ational early-warning system. *Ocean Dynamics* 62, 1001–1015.
- 568 Wolpert, D.H., 2002. The supervised learning no-free-lunch theorems, in: *Soft Com-*
569 *puting and Industry*. Springer, pp. 25–42.