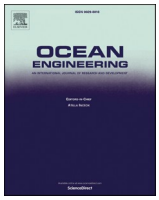
Ocean Engineering 236 (2021) 108699

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Improving accuracy on wave height estimation through machine learning techniques

S. Gracia [a], J. Olivito [a,*], J. Resano [a], B. Martin-del-Brio [a], M. de Alfonso [b], E. Álvarez [b]

[a] University of Zaragoza, Spain
[b] Puertos Del Estado, Spain

ARTICLE INFO

Keywords:
Wave height
Machine learning
Neural network
LightGBM

ABSTRACT

Estimatabion of wave agitation plays a key role in predicting natural disasters, path optimization and secure harbor operation. The Spanish agency Puertos del Estado (PdE) has several oceanographic measure networks equipped with sensors for different physical variables, and manages forecast systems involving numerical models. In recent years, there is a growing interest in wave parameter estimation by using machine learning models due to the large amount of oceanographic data available for training, as well as its proven efficacy in estimating physical variables.

In this study, we propose to use machine learning techniques to improve the accuracy of the current forecast system of PdE. We have focused on four physical wave variables: spectral significant height, mean spectral period, peak period and mean direction of origin. Two different machine learning models have been explored: multilayer perceptron and gradient boosting decision trees, as well as ensemble methods that combine both models. These models reduce the error of the predictions of the numerical model by 36% on average, demonstrating the potential gains of combining machine learning and numerical models.

1. Introduction

The state-owned Spanish Port System includes 46 ports of general interest, managed by 28 Port Authorities, whose coordination and efficiency control corresponds to the government agency Puertos del Estado (PdE), that is responsible for implementing the government's port policy. Accurate estimations of wave parameters (height, period and direction), both in the open sea and in port areas, are particularly important for several reasons. Estimations on high seas allow predicting dangerous phenomena or events caused by natural catastrophes (Vanem, 2011; Dixit and Londhe, 2016). In terms of logistics, these estimations make it possible to optimize routes for vessels, increasing safety and cost savings (Zheng and Sun, 2016; Liu et al., 2016).

The energy sector is another field greatly benefited by achieving more accurate estimations. In the last few years, the number of wave-based energy generation systems (wave energy converters) has increased considerably (López et al., 2013; Bahaj, 2011; Falcão, 2010). In order to determine the viability and productive capacity of this type of systems, it is essential to know with great precision the wave history in an area in order to predict future trends (Cuadra et al., 2016). Speaking purely from the oceanographic and climatic point of view, having a reliable wave history makes it possible to carry out valid analysis and detect changes in trends.

Wave parameters estimation in port areas is of vital importance for security reasons, allowing the loading and unloading of goods to be carried out in safe conditions, and permitting the corresponding port authority to determine if the port should be closed at the entrance of ships, causing certain vessels to divert to nearby ports. Wave parameters estimation has been traditionally carried out by means of numerical models forced with wind fields that reproduce the processes of wave generation and propagation. The output of these models are gridded wave spectra, from which wave parameters are estimated. These models should be validated and calibrated with real measurements that in most cases are provided by wave sensors included in buoys. These buoys have been measuring during decades and there is an increasing amount of historical qualified wave dataset. However, there are some periods when the buoys could not provide information due to malfunctions or drifts generating gaps in the historical dataset. The numerical models, forced with historical wind fields can provide hindcast datasets without gaps but differing from the real measurements. These errors can be due to

* Corresponding author.
E-mail addresses: sergio90gb@gmail.com (S. Gracia), jolivito@unizar.es (J. Olivito), jresano@unizar.es (J. Resano), bmb@unizar.es (B. Martin-del-Brio), mar@puertos.es (M. de Alfonso), enriquemar@puertos.es (E. Álvarez).

https://doi.org/10.1016/j.oceaneng.2021.108699
Received 30 June 2020; Received in revised form 11 December 2020; Accepted 31 January 2021
Available online 30 August 2021

Table 1
Input and output variables in the data sets used in our machine learning models.

	Data source	Variables		
		Acronym	*Description*	*Unit*
Inputs	*Date*	yyyy/mm/dd	Date	year/month/day
	Time	hh	Time of the day	hour
	Wave agitation model	Hm0	Significant spectral height	meters (m)
		Tm02	Spectral mean period	seconds (s)
		Tp	Spectral peak period	seconds (s)
		DirM	Mean direction of wave origin	degrees (°)
	Wind model	VelV	Wind mean speed	meters per second (m/s)
		DirV	Mean direction of wind origin	degrees (°)
	Water current model	uo	North component of mean current speed	meters per second (m/s)
		vo	East component of mean current speed	meters per second (m/s)
Output Evaluation	*Buoy*	Hm0	Significant spectral height	meters (m)
		Tm02	Spectral mean period	seconds (s)
		Tp	Spectral peak period	seconds (s)
		DirM	Mean direction of wave origin	degrees (°)

different sources: inaccuracies in the bathymetry, errors in the input wind field or errors in the generation or propagation of wave energy from the wave model. Models based on machine learning can be used to reduce these errors due to their ability to find correlations in the input data ("learning from data" models), however complex it may be. In this work, we have developed machine learning models (Marsland, 2009; Géron, 2020) that improve accuracy on wave height parameter estimations from numerical models at different locations of the Spanish coastline. After exploring different machine learning techniques, two different models have been selected: Multilayer perceptron (MLP) and gradient boosting decision trees (GBDT), and ensemble methods that combine both have also been analyzed. As inputs for our machine learning models we have used the estimations of the numerical model currently used by PdE and a set of variables recorded by their buoy network enumerated in Table 1. With this information our machine learning models have updated the predictions of the numerical model in order to improve its accuracy.

The paper is structured as follows. Section 2 presents the related work. Section 3 outlines the machine learning models selected for this study. Section 4 describes the data used in the study and the preprocess carried out. Section 5 presents the experimental results. Finally, Section 6 summarizes our conclusions.

2. Related work

The estimation of oceanographic variables through machine learning models has been the target of many research works in the last years, hand in hand with the renewed emergence of artificial intelligence models based on "learning by data". Different approaches have been proposed depending on the source data used to build the prediction model. A very widespread method is the use of data from adjacent deep-water buoys to estimate the measurements of a target buoy (López et al., 2015; Alexandre et al., 2018; Cornejo-Bueno et al., Salcedo-Sanz; Krishna-kumar et al., 2017; Mahjoobi and Mossabeb, 2009; Etemad--Shahidi and Mahjoobi, 2009). This method allows providing wave parameter estimations at the location of the target buoy even when it is out of service. One advantage of this method is that it does not require use of numerical models, which are usually very computationally demanding. Its main limitation is its dependence on the instrumentation availability, noticeable lower than data coming from numerical models. In (Mahjoobi and Mossabeb, 2009) and (Etemad-Shahidi and Mahjoobi, 2009) the authors propose to use regressive support vector machines and MLP, respectively, for wave height predictions. In (Alexandre et al., 2018) the authors propose to combine genetic algorithms with machine learning models to locally reconstruct the output of out-of-operation buoys in the Caribbean Sea and West Atlantic (Cornejo-Bueno et al., Salcedo-Sanz). continues the work of (Alexandre et al., 2018) but includes a Bayesian Optimization method to select the attributes used by their prediction model. In (Krishna-kumar et al., 2017) the authors propose to use sequential learning algorithms, namely the Minimal Resource Allocation Network (MRAN) and the Growing and Pruning Radial Basis Function (GAP-RBF) network, to predict the daily wave heights in different geographical regions. In (López et al., 2015) the authors propose the use of MLPs to estimate wave agitation within a port basin based on deep-water observations alone. Specifically, this work is based on the estimation of the wave height in the buoy of the port of Ferrol (La Coruña, Spain) from data from two adjacent buoys.

A second approach consists in predicting future wave agitation based on the last measurements recorded by a target buoy (Yin et al., 2013; Pashova and Popova, 2011; Yasseri et al., 2010). (Yin et al., 2013) presents the application of a sequential learning radial basis function network for real-time prediction of tidal level (Pashova and Popova, 2011). predicts the daily mean sea levels in the Black Sea coast using MLPs (Yasseri et al., 2010). predicts the significant wave height and mean zero-up-crossing wave period in the north east Pacific using MLPs and a finite element method.

A third approach of posing the problem is to consider partially or fully replacing a previous numerical model. For example, in (Puscasu, 2014) the authors propose to use an MLP to approximate the result of the resolution of the non-linear term (Snl) to reduce the computation time of numerical models. Other works propose to fully replace the numerical model by a machine learning model (Malekmohamadi et al., 2008; Pooja et al., 2011; James et al., 2018). In (Malekmohamadi et al., 2008; Pooja et al., 2011) this is done for a specific buoy using an MLP (Malekmohamadi et al., 2008), and a model that combines MLP, genetic programming and model tree (Pooja et al., 2011). (James et al., 2018) pursues a more ambitious goal and recreates the estimations of the SWAN (Scientific and techn, 2009) numerical model by using MLPs for the Bay of Monterey (USA). The MLPs predictions are 4,000 times faster than the SWAN model, and the root mean square error (RMSE) was less than 5% of the mean value of the swell. The authors explain that the Bay of Monterey has stable conditions that allow reaching these results. Trying to develop methods to replace numerical models in more exposed areas, and with more changing conditions, is still a hard challenge.

Another option is to consider machine learning as a complementary tool to improve the estimations of numerical models. Predictions made by the numerical models are post-processed to bring them closer to the real measurements obtained by a buoy at a given point, at the expense of minimally increasing the computational cost (Makarynskyy, 2004; Makarynskyy, 2006; Zhang et al., 2006; Filippo et al., 2012; LightGBM examples, 2020). (Makarynskyy, 2004) proposes to use MLPs to update the prediction of weight height (Makarynskyy, 2006). also propose an MLP model to update the predictions for wave height, zero-up-crossing wave period and peak wave period. They use the data of a buoy station coupled with a numerical model to predict the wave conditions in two points close to the buoy station. In their experiments they reduce the root mean square error from 2.2 to 3.9 (Zhang et al., 2006). uses MLPs to improve the predictions in four points of the northwestern Pacific Ocean, reporting a consistent improvement in the predictions (Filippo et al., 2012). analyzes the case of the Cananéia coasts (Brazil) reducing the error of the numerical model from 26% to 12% by using MLPS.

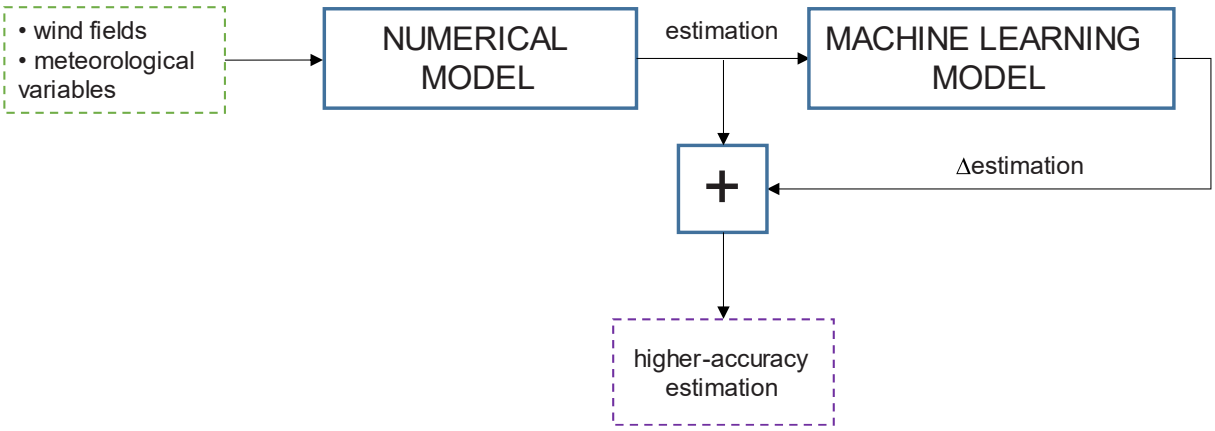


Fig. 1. Inference workflow to increase the accuracy of numerical model predictions through machine learning.

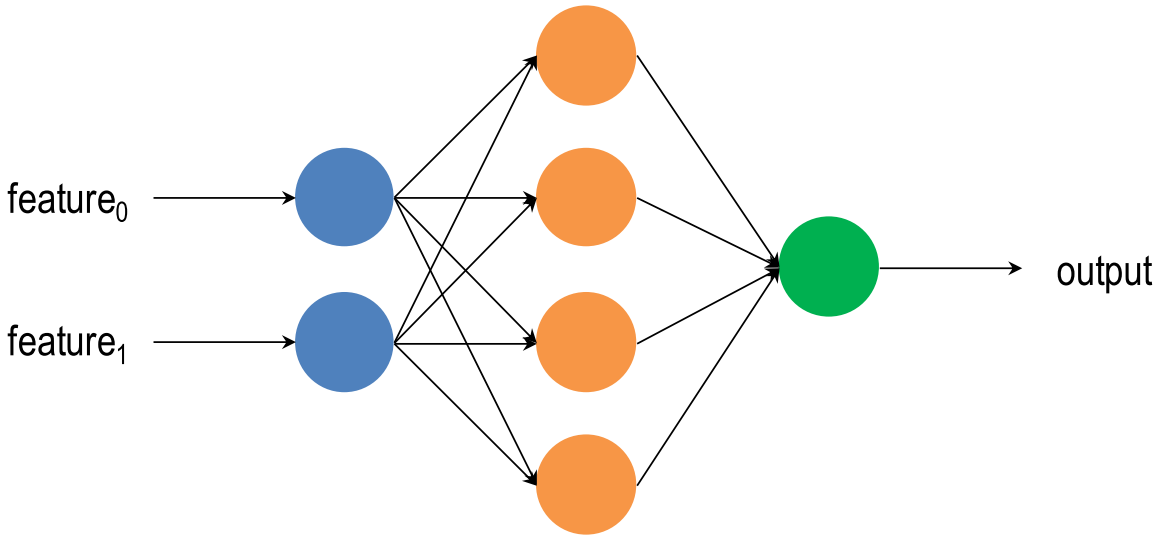


Fig. 2. Example of a 2-4-1 MLP architecture. Two input neurons send two features to a hidden layer including four neurons, and one output neuron carries out the final computations and generates the output.

Finally (Ellesonet al., 2020) is a very recent reference in which the authors use bagged decision trees to detect deviations of the wave height in the predictions of the numerical wave model (WaveWatch III) using its outputs and wind information from a forecast system. When these deviations were applied as corrections error metrics root-mean-squared-error, bias, percent error, and scatter index were reduced in several different experiments. Moreover they developed a descriptive tool that identifies regions with similar errors.

Our work follows a similar approach to those described in the previous paragraph: apply machine learning techniques to improve the results of a numerical model. First, classical MLPs will be used for two reasons, because they are universal approximators (Goodfellow et al., 2016), and because they are used with good results in many of the previous works found in the literature (as it can be seen in the above paragraph). Second, we will use decision trees as proposed in (Ellesonet al., 2020). Nevertheless, other modern machine learning techniques have also been explored. Thus, we have identified that Gradient Boosting Decision Trees (GBDT) is a powerful technique that achieves better results than MLPs or traditional decision trees for our predictions. Moreover, two additional ensemble techniques have been applied to further improve the results: bootstrap aggregating (bagging) (Breiman, 1996) of several MLP and combining the results of MLP bagging and the GDBT. With this combination slightly better results and a more robust model are obtained. In order to validate our approach, our machine learning models have been evaluated in four locations on the Spanish coast that are exposed to very different sea state conditions. The benefits of our approach vary from one point to another, but even in the most complex cases, our machine learning models can improve the predictions of some of the physical wave variables.

3. Machine learning models for wave height estimation

In this section, the Machine learning (ML) models used in our work will be described. ML models (Marsland, 2009) are data-driven techniques that automatically learn patterns and input-output relationships from data sets. Thus, they are suitable to tackle the prediction of physical wave variables because they are very efficient in detecting patterns and complex relationships between input data, in order to estimate the value of an output variable. In addition, in our case, a large amount of historical data is available for training machine learning models.

In this work, the two-stage scheme shown in Fig. 1 is proposed. The Spanish agency Puertos del Estado (PdE) has several measure networks equipped with sensors for different physical variables, and forecast systems based on numerical models. In our approach, the machine learning model receives the output predictions provided from the numerical model, and estimates a correction which leads, on average, to a more accurate prediction of physical wave variables (Section 5).

Among the collection of models available in the machine learning

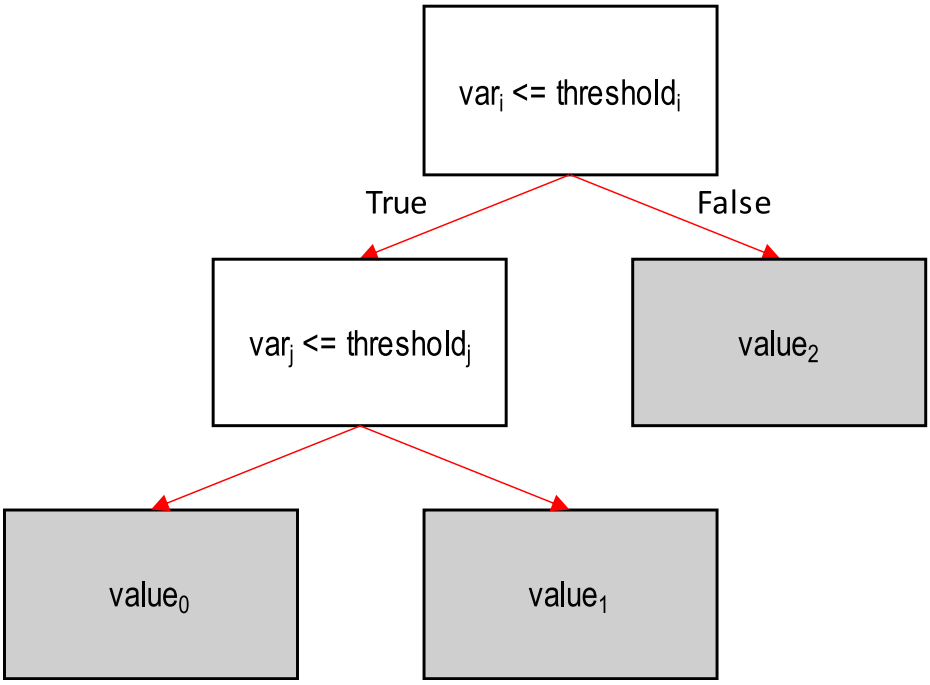
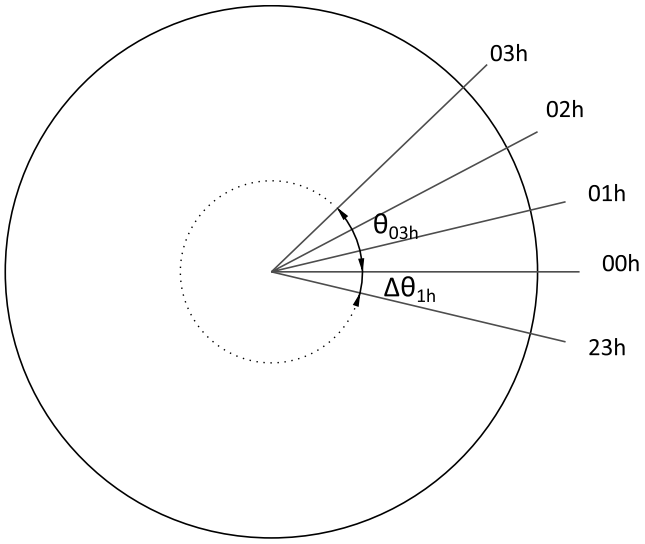


Fig. 3. Example of a decision tree. White boxes represent conditions to be evaluated (split points, feature values depending on which data is divided at a tree node), and gray boxes are leaves nodes (containing the output values).



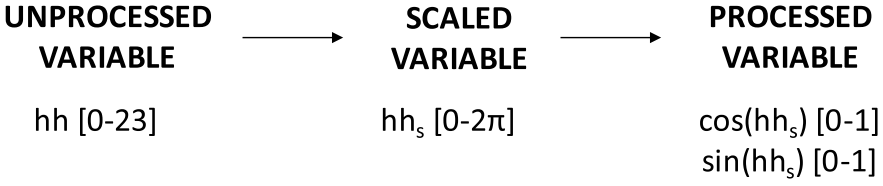


Fig. 4. Standardization on cyclic variables.

ecosystem (Marsland, 2009; Goodfellow et al., 2016; Haykin, 1999), neural networks (Section 3.1) and decision trees (Section 3.2) have been selected. In addition, ensemble techniques to further increase accuracy have also been evaluated (Section 3.3).

For all our models, a time-series cross-validation strategy is applied, where the corresponding training set consists only of samples that occurred prior to the samples that form the test set, thus, no future data is used in constructing the forecast system. For instance, consider a

Fig. 5. Physical locations of the buoys. From North to South: Villano, Tarragona, Tarifa and Tenerife.

temporally ordered dataset divided in four subsets, A, B, C and D, the procedure starts by training with subset A and testing with B (fold 1), it continues by training with A and B, and testing with C (fold 2), and the procedure ends by training with A, B, C, and testing with D (fold 3).

All these machine learning models are complex and involve powerful mathematical developments, thus, in next sections we only explain the basic ideas of these algorithms; the interested reader can find more algorithmic and mathematical details, for instance, in references (Marsland, 2009; Goodfellow et al., 2016; Géron, 2020; Haykin, 1999).

3.1. Multi-layer perceptron (MLP)

Multi-layer perceptrons (MLP) are one of the most widely-used artificial neural network (ANN) architectures (Haykin, 1999; Marsland, 2009; Géron, 2020) because they are universal approximators (Goodfellow et al., 2016; Haykin, 1999) that are able to capture nonlinear input-output relationships from a data set. Specifically, the universal approximation theorem, developed by Cybenko and Hornik (Goodfellow et al., 2016), states that a feed-forward neural network with at least one hidden layer can approximate any multivariate function, provided that the network is given enough hidden units. MLPs were used in most of the works presented in Section 2.

An MLP (Fig. 2) consist of an input layer, with as many neurons as input features; hidden layers, whose neurons compute a weighted sum of its inputs, and then apply a nonlinear function, typically a sigmoid function or a rectified linear unit (ReLU) (Géron, 2020); and an output layer, that generates the final output by computing the weighted sum of the last hidden layer outputs. In an MLP, each neuron is connected to every single neuron of the previous layer, and each connection is modeled with a weight. These weights are iteratively adjusted from the dataset by using an optimization algorithm, which minimizes a cost function that compares the current MLP output versus the desired one. MLPs are often trained following the stochastic gradient descent algorithm, where the gradient is usually computed by using the back-propagation technique. Algorithmic and mathematical details can be found in references (Marsland, 2009; Haykin, 1999).

3.2. Decision trees

A decision tree (Marsland, 2009) (Fig. 3) is a machine learning model that uses an if/then/else branching method to represent every possible model output. Thus, it consists of a set of if/then/else conditions (split points) about each input feature in turn, starting at the root of the tree and progressing down to the leaves where the corresponding final output is assigned (Marsland, 2009). In short, decision trees use a tree-like model of chained if/then/else decisions and their possible consequences for achieving the final model output.

Fig. 3 shows a scheme of a binary decision tree. White nodes represent conditions, where an input feature or variable is compared with a threshold (split point), thus determining the subsequent path depending on the true or false result of the evaluated condition. Nodes in gray are leaves nodes, which are labeled with the corresponding output value. Thus, the tree is traversed (top-down) by comparing at each level one input feature with a threshold (adjusted during training) until a leaf node (which contains the prediction) is reached. In the worst case it requires depth -1 comparisons, but, as can be seen in Fig. 3, the output value can be found at any level.

The training algorithm determines which feature is used at each level and the numeric value of the thresholds used in every split point, in a way that minimizes the mean squared error (MSE); one of the most popular algorithms for this purpose is the Classification and Regression Tree (CART) algorithm (Timofeev, 2004).

3.3. Ensemble learning

Ensemble learning (Marsland, 2009) is a machine learning paradigm where several machine learning models are trained to solve the same problem and combined to get better results. When several models are correctly combined, a more accurate or robust model can be obtained. Thus, ensemble learning (Marsland, 2009) combines predictions coming from different models in order to improve the final prediction. In our experiments we have used three different ensemble methods: gradient boosting, bagging and averaging.

Models based on decision trees usually rely on the prediction as a result of many single estimators (i.e. decision tree). Once trained, the individual trees are combined by using ensemble methods; a typical approach for this are random forest techniques (Breiman, 2001). The final output is achieved by computing the mode (for classification) or the mean prediction (for regression) of the individual trees. Thus, by combining several predictions, a stronger predictor is obtained.

However, even better results can be obtained, by applying an ensemble approach called gradient boosting (Géron, 2020). Gradient boosting works sequentially by adding the outputs of several decision trees (predictors), each one correcting its predecessor, in such a way that each tree attempts to improve the results of the previous. Every predictor is trained sequentially, so each new iteration tries to correct the residual error generated in the previous one. Once the trees are trained, they can be used for prediction by simply adding the outputs of all the trees (Géron, 2020). For instance, consider an ensemble of three decision trees (DT). DT number one (DT1) is trained normally, providing an output $h1$ with some residual error (actual output minus predicted output). Then DT2 is trained on the residual error of DT1 (now residual errors are target values); DT2 output $h2$ (a correction to DT1 output) is added to that of DT1 for obtaining the ensemble output h, $h = h1+h2$, thus reducing the error provided by DT1. Finally, DT3 is trained on the residual error of DT2 and its output $h3$ is added for obtaining the final ensemble output $h = h1+h2+h3$, reducing the error even more.

Gradient boosting is used by Gradient Boosting Decision Trees (GBDT) (Jerome, 2002). Conventional implementations of GBDT suffer from poor scaling for large datasets or a large number of features. LightGBM (Guolin et al., 2017) is a highly efficient open-source GBDT-based framework that overcomes this drawback by excluding a significant proportion of data instances with small gradients, and by bundling mutually exclusive features (that rarely take nonzero values simultaneously), thus offering up to 20 times higher performance over conventional GBDT. For this reason, we have selected LightGBM in our work. With the support of LightGBM, GDBTs are currently considered one of the most powerful machine learning models due to its efficiency,

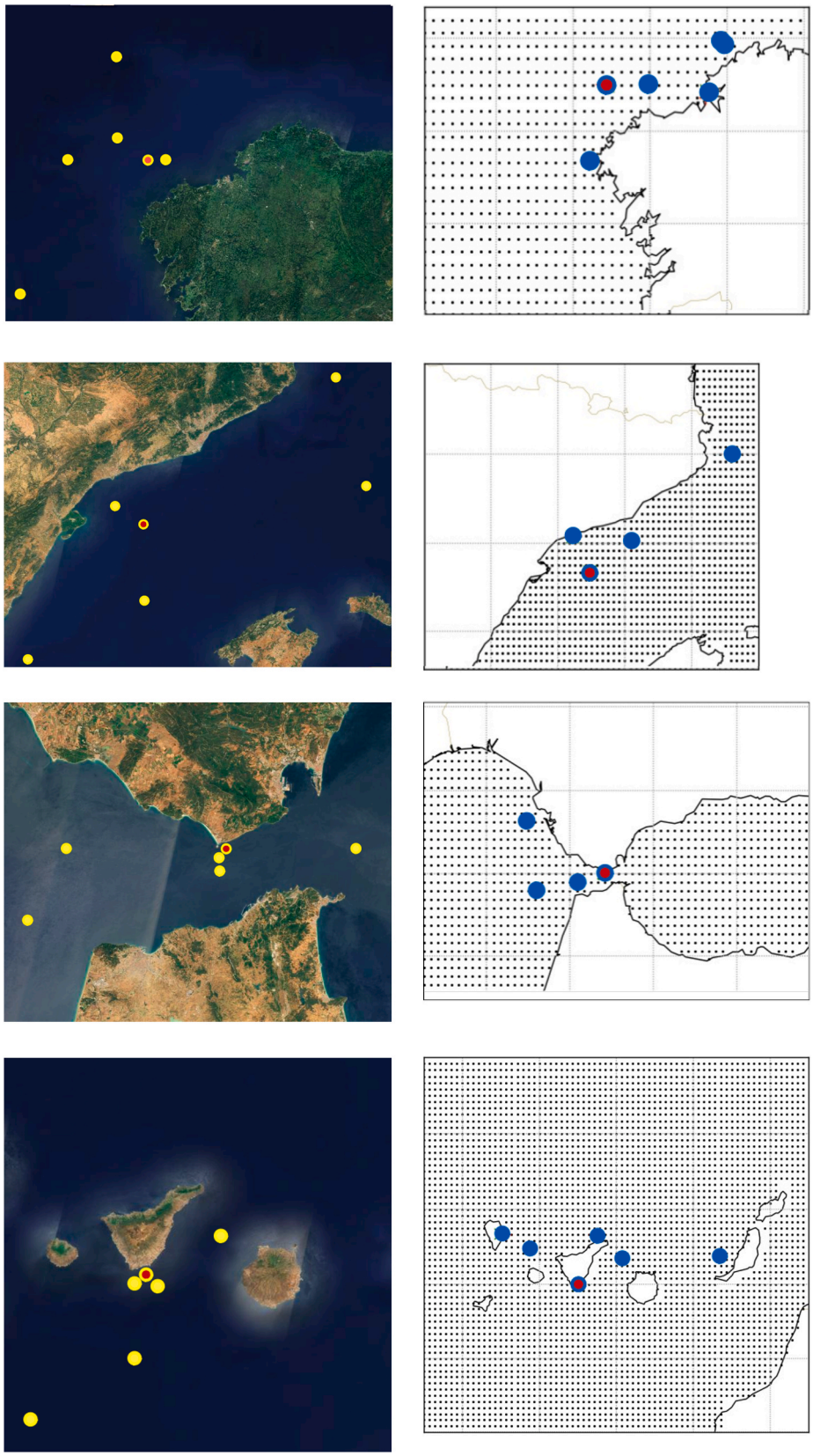


Fig. 6. Physical locations of the model points and buoys used to develop our ML models. Rows: Villano, Tarragona, Tarifa and Tenerife. Columns: waves-winds model points (yellow) and currents model points (blue). The points filled in red show the spatial point where a buoy and a model point are overlapped.

Table 2
Coordinates of the model points and buoys. Points for each location/model are sorted by latitude (from north to south) and by longitude (from west to east).The rows in bold show the spatial point where a buoy and a model point are overlapped.

Location	Waves-winds model		Currents model	
	Latitude (°)	Longitude (°)	Latitude (°)	Longitude (°)
Villano	44.250	−9.500	43.751	−8.166
	43.667	−9.500	43.501	−8.833
	43.500	−10.000	43.501	−8.166
	43.500	**−9.208**	43.417	−8.333
	43.500	−9.000	43.001	−9.333
	42.500	−10.500		
Tarragona	41.917	3.667	42.000	3.584
	41.000	4.000	41.084	1.251
	40.833	1.167	41.000	2.084
	40.667	**1.500**	40.667	1.501
	40.000	1.500		
	39.500	0.167		
Tarifa	36.000	−6.000	36.500	−6.500
	36.000	−5.583	36.000	−5.583
	36.000	−5.250	35.917	−5.916
	35.983	**−5.600**	35.834	−6.416
	35.950	−5.600		
	35.850	−6.100		
Tenerife	28.250	−16.000	28.667	−17.667
	28.000	**−16.583**	26.667	−16.333
	27.917	−16.667	28.500	−17.333
	27.917	−16.500	28.417	−14.417
	27.417	−16.667	28.333	−15.917
	27.000	−17.500	28.000	−16.583

accuracy and interpretability. For example, recently they have been used for many winning solutions in several machine learning competitions (LightGBM examples, 2020).

Bootstrap aggregating (bagging) (Breiman, 1996) builds multiple estimators by training each of them with a subset of the training data, and then aggregates their predictions. These subsets are built by picking random samples with replacement. We have applied bagging to improve the results obtained by our MLP models and, in addition, bagging is also used in the core of the LightGBM algorithm.

Finally, averaging is a simple approach to combine different machine learning models for regression. Each model is trained independently, and during inference their outputs are averaged to generate the final results.

In summary, four different approaches have been evaluated:

1) Single MLP: This is our simplest ML model. Best hyperparameters for each location are found by random search. In order to mitigate the effect of model weight initialization, results shown in this category are the average error and training time for 50 instances of a single MLP.
2) Bagging MLP: Based on the optimal hyperparameters previously found, we consider a bagging of N MLPs, where each one is trained on a random fraction M of the training set. Hyperparameters N and M are adjusted for each location as shown in Table 3.
3) LightGBM: As indicated in Section 3, LightGBM is an efficient implementation of GBDT (Gradient boosting decision trees). It consists of N single estimators (decision trees) where each tree attempts to reduce the residual error of the previous one. Each tree is trained on a random fraction M of the training set (thus, LightGBM algorithm also uses bagging). M, N, and other hyperparameters are tuned for each location by random searching.
4) Ensemble: Predictions coming from a bagging of MLPs and a LightGBM are averaged.

4. Methods

This section presents the data sets used in our experiments, the preprocessing steps needed before using the data, the field sites selected to evaluate our model, and the description of the experiments carried out.

4.1. Data sets and preprocessing

Puertos del Estado (PdE) provided us with data sets for each location on the Spanish coast. Each data set includes their numerical models estimations (for wave height, wind, and water current) and the measurements recorded by their buoy network. The time resolution of these data sets is one sample per hour. Table 1 enumerates the variables used as inputs for our models, which includes the date and time, as well as the outputs of three different computational models. It also includes the additional information used to evaluate the outputs of our model, which are measurements taken at the buoys.

In order to preprocess data for the machine learning models, we applied two standardization methods on raw data depending on the nature of the variable. In the case of acyclic variables (*Hm0*, *Tm02*, *Tp*, *VelV*, *uo* and *vo*), for putting all the variables in the same scale, they are standardized according to

$$z = \frac{x - \underline{x}}{S} \tag{1}$$

where z is the standardized value, x is the raw value, $\underline{x}$ is the mean in the training set, and S is the standard deviation in the training set. As it is well known, equation (1) transforms a variable x into another variable z with mean 0 and standard deviation 1.

Cyclic variables demand a different approach to properly reflect their underlying proximity in time or space. In our dataset, cyclic variables are timestamps - where we considered both the day within a year and the time within a day -, and directional variables (*DirM* and *DirV*). Fig. 4 illustrates standardization on the time within a day. Unprocessed variable *hh* is first scaled to $[0–2\pi]$, and then sine and cosine of this scaled value are computed and presented as processed variables. The purpose of this method is to force that hours close in time are also close in value (for example, 01h and 23h).

4.2. Field Sites

Our machine learning approach has been evaluated in four locations

Table 3
Datasets on each location.

Location	Wave & wind model points	Current model points	Total input variables	Training set		Test set	
				Period	#samples	Period	#samples
Villano	6	5	295	2005–01 2014–08	73,266	2014–09 2017–03	18,317
Tarragona	6	4	285	2005–01 2014–12	73,722	2015–01 2017–03	18,431
Tarifa	6	4	285	2009–01 2016–01	45,997	2016–02 2017–06	11,499
Tenerife	6	6	305	2005–01 2014–08	77,987	2014–09 2017–03	19,496

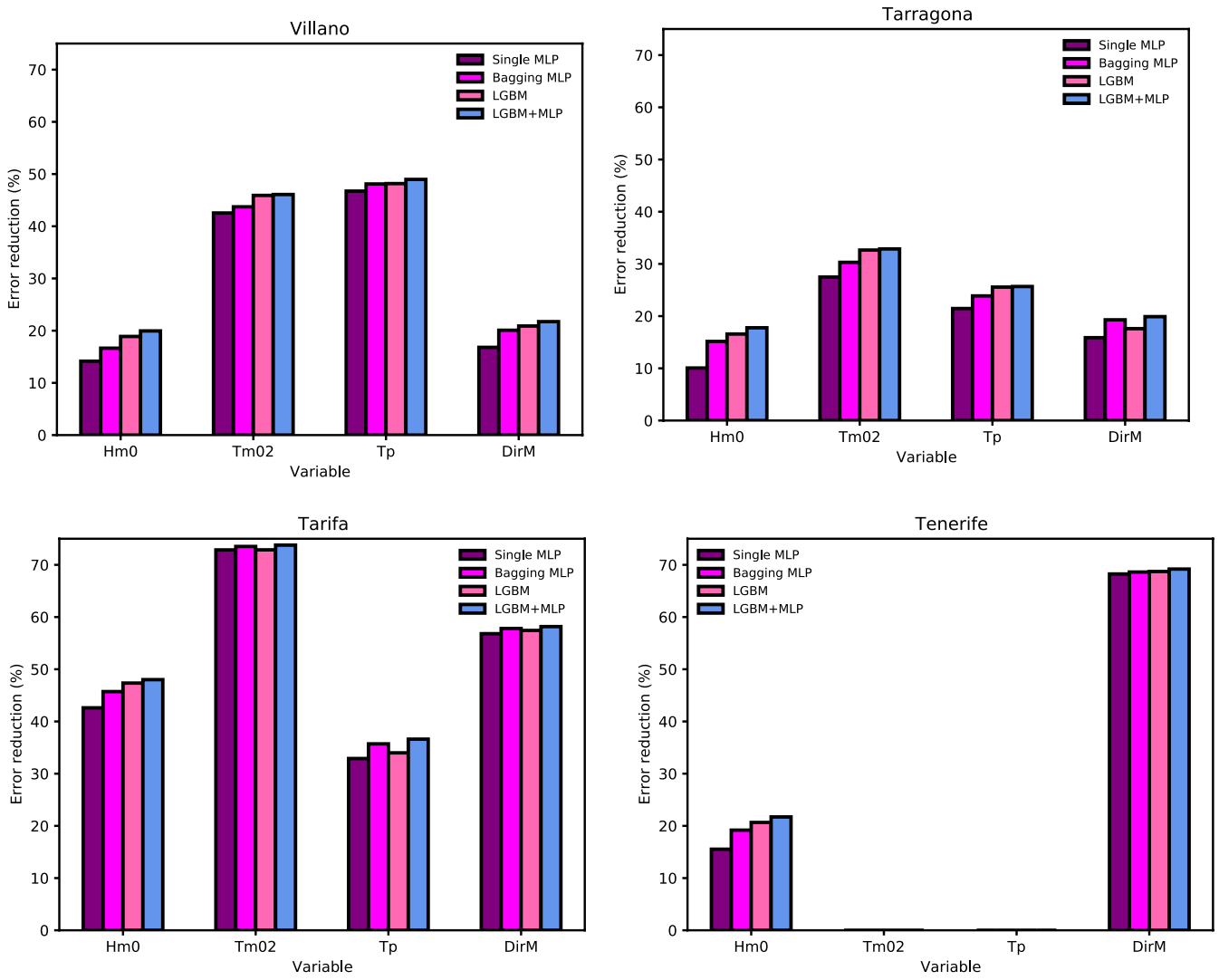


Fig. 7. Error reduction for each location and variable achieved by our ML models in the test sets.

on the Spanish coast: Villano, Tarragona, Tarifa and Tenerife (see Fig. 5). These locations have been selected as they are exposed to very different sea state conditions for testing our methodology: Villano is in the Atlantic Coast, Tarragona in the Mediterranean Sea, Tarifa in the Gibraltar Strait and Tenerife in the Canary Island.

In order to make estimations in each buoy location, PdE selected key points from their wave height and current numerical models for each location. The model points have been chosen trying to be the most representative in the grid, providing all the possible contributions that can affect the results in every location: the closest points to the buoy and the tide gauge, and those in strategic positions for oceanographic phenomena. For Villano position, apart from the closest point to the buoy, the selected points were three outer points able to represent the wind and wave contributions from the North, South and North-West directions and four points to represent the Iberian Poleward current. In Tarragona, The selected points are representative of the North and South contributions for wave and wind and representative of the typical slope current in the Catalonian Coast from the North. In Tarifa position, the points chosen are representative of the easterlies and westerlies main wind and wave regimes and the eastward surface main current in the Gibraltar Strait. For Tenerife, the selected points are the most representative of the North and South wind and wave contributions that affect the buoy area and the representative of the main currents in the Canary Island. The physical locations of these points are shown in Fig. 6; waves and wind model points on the left, and physical location of the currents model points on the right (buoys are shown as red points). Specific coordinates for each point are shown in Table 2.

4.3. Experiments

We built our regression models as a function of not only the current sample but also the four previous ones. Therefore, a valid sample in our model requires five consecutive raw samples with data for all variables available.

Table 3 characterizes the dataset used on each location; all ML models were trained with wave, wind and current values provided by the numerical models. The whole dataset was split into a training set (80%) and a test set (20%). The field 'Total input variables' is the aggregation of the input variables (see Table 1) for each model point for the last 5 h. For example, for Villano it includes 1 input for the year, 4 inputs for the normalized time and date, 240 inputs for the six points of the wave and wind model during the last 5 h ($6 \times 5 \times 8$), and 50 inputs for the five points of the current model during the last 5 h($5 \times 5 \times 2$).

For each sample our model generates for different outputs. However these outputs are not the absolute values that we want to predict (Hm0, Tm02, Tp, DirM, described in Table 1), but, following the

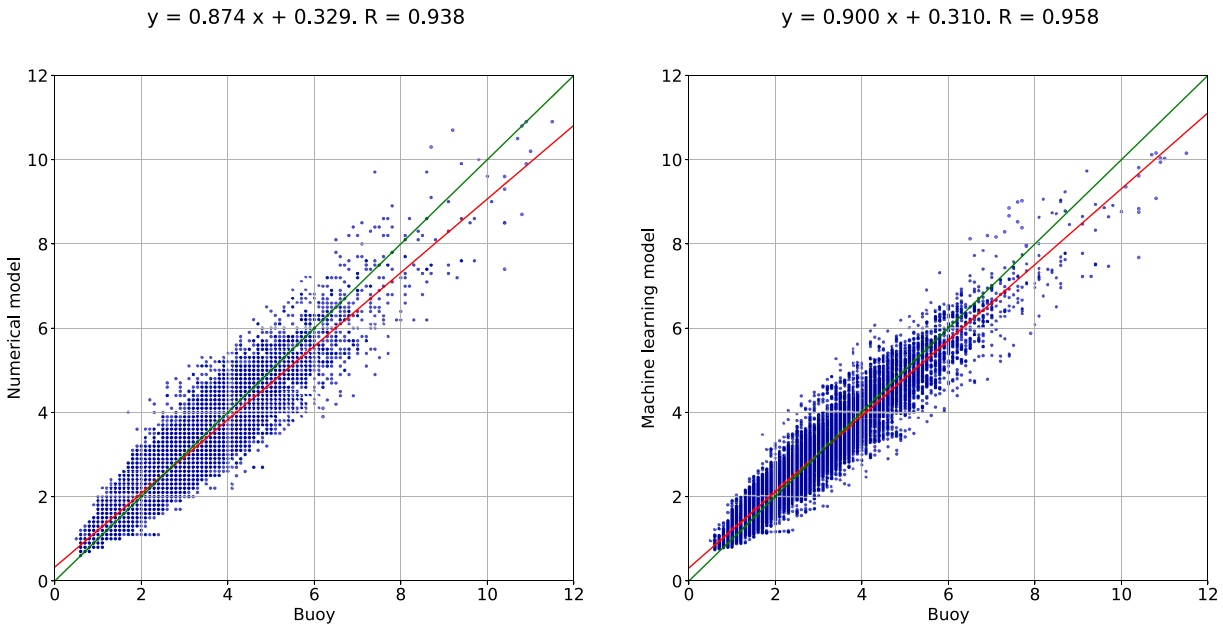


Fig. 8. Scatter plot for Villano - *Hm0*.

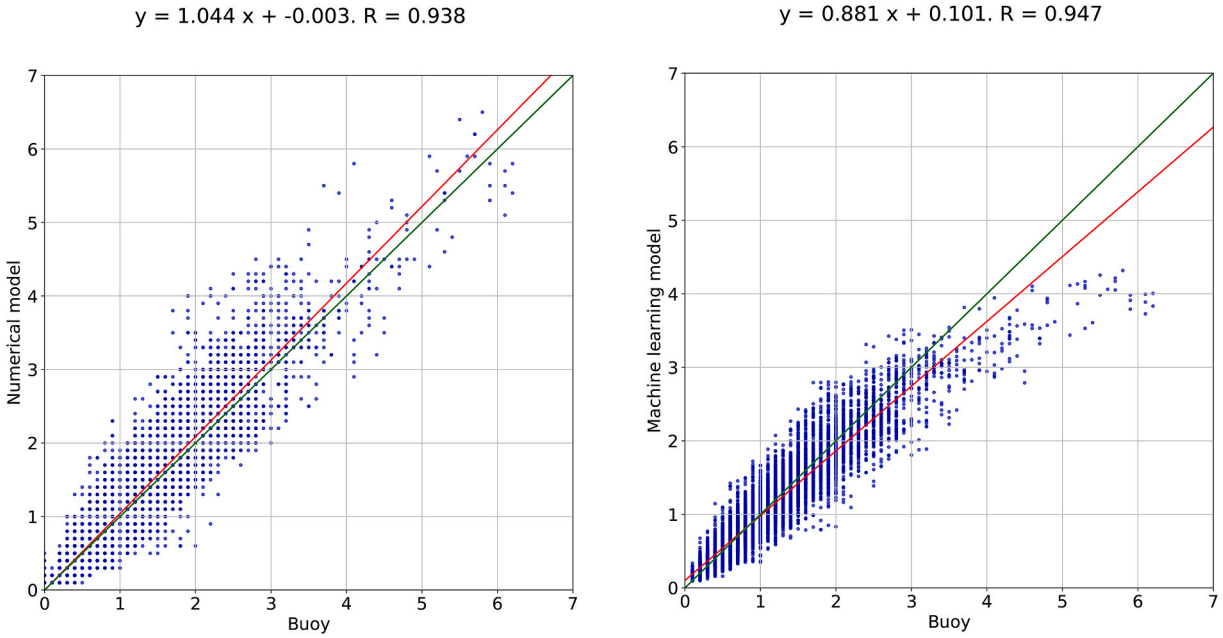


Fig. 9. Scatter plot for Tarragona - *Hm0*.

recommendation of (KantardzicData Mining: Concepts et al., 2011), our models generate a correction that must be applied to the initial prediction obtained with the numerical model. For acyclic variables measured by the buoy, *Hm0*, *Tm02* and *Tp*, the objective of the machine learning model is to estimate the increase Δ that corrects the wave model estimation (Δ*Hm0*, Δ*Tm02* and Δ*Tp*), as shown in (2)

$$\Delta output\_var = output\_var_{buoy} - ouput\_var_{numerical\_model} \quad (2)$$

For angular output variables (i.e., *DirM*), we decided to estimate the correction of the sine and cosine (Δsin and Δcos) and then reconstruct the angle by computing the arc tangent.

$$angle_{ML\_model} = \arctan(sine_{ML\_model} / cosine_{ML\_model}) \quad (3)$$

Notice that this method yields predictions where the trigonometric relationship $\sin^2 \alpha + \cos^2 \alpha = 1$ is not guaranteed. However, it provides the best results as it represents a kind of ensemble by mixing the independent sine and cosine predictions.

For each model output we compute the original error of the numerical model as:

$$output\_var\_numerical\_model\_error = | output\_var_{buoy} - ouput\_var_{numerical\_model} | \quad (4)$$

And the error after including our ML model to correct the initial results as:

$$output\_var\_ML\_error = |output\_var_{buoy} - (ouput\_var_{numerical\_model} + ouput\_var_{ML\_model})| \quad (5)$$

Comparing these two errors we identify the error reductions due to

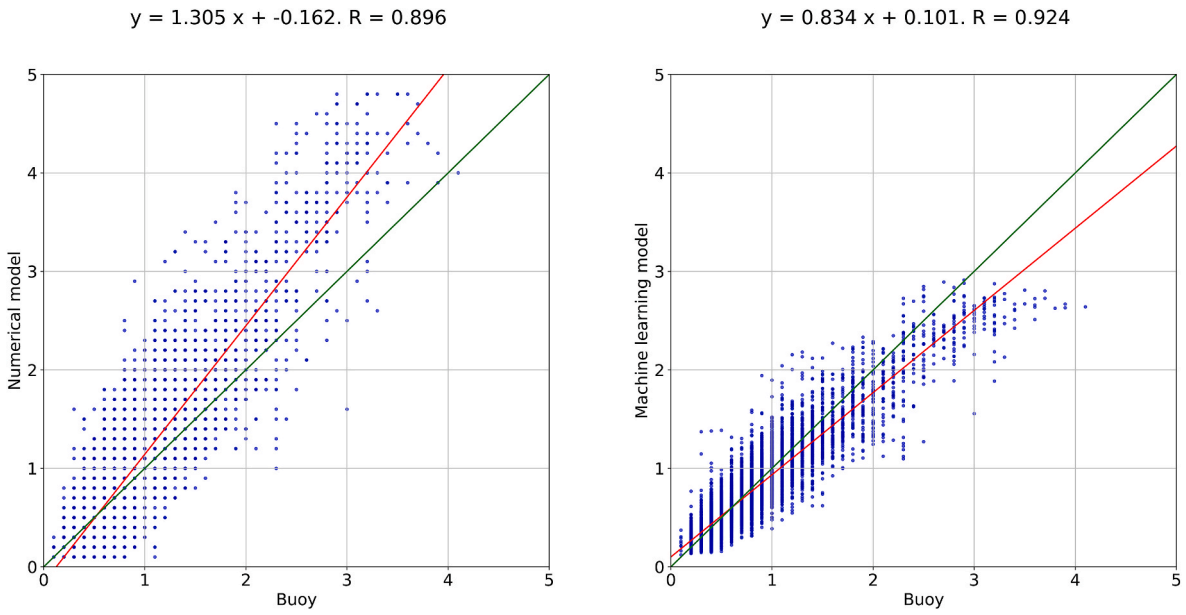


Fig. 10. Scatter plot for Tarifa - *Hm0*.

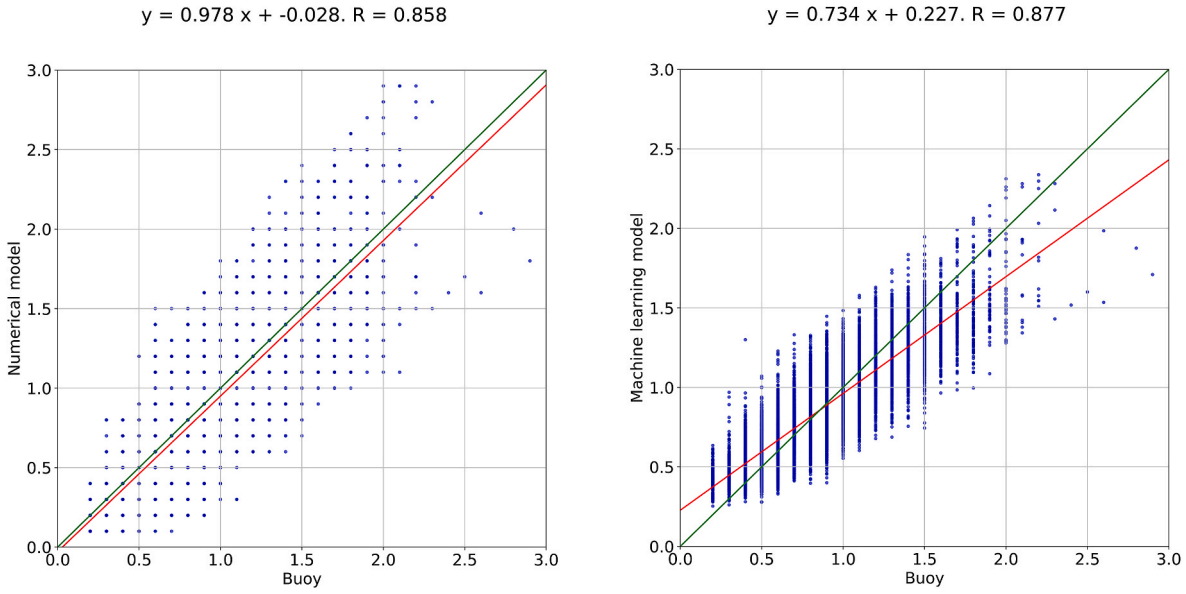


Fig. 11. Scatter plot for Tenerife - *Hm0*.

Table 4
Mean absolute error (MAE) and correlation coefficient (R) on the test set for each location.

	Villano				Tarragona				Tarifa				Tenerife			
	Numerical model		ML model		Numerical model		ML model		Numerical model		ML model		Numerical model		ML model	
	MAE	R	MAE	R	MAE	R	MAE	R	MAE	R	MAE	R	MAE	R	MAE	R
Hm0 (m)	0.33	0.94	0.27	0.96	0.17	0.94	0.14	0.95	0.23	0.90	0.12	0.93	0.16	0.86	0.12	0.87
Tm02 (s)	0.71	0.90	0.38	0.93	0.41	0.87	0.28	0.90	1.56	0.56	0.41	0.79	–	–	–	–
Tp (s)	1.611	0.83	0.82	0.86	0.94	0.70	0.70	0.77	1.61	0.43	1.03	0.70	–	–	–	–
DirM (°)	11.54	–	9.60	–	27.28	–	24.26	–	39.24	–	17.9	–	45.44	–	13.61	–

the corrections carried out by our models.

5. - Experimental results

Fig. 7 shows the accuracy improvement over the numerical model achieved by our four approaches for each location and output variable in the test sets. The first column presents the results when using a single MLP, which is the technique used in most of the previous references (Section 2). This approach yields error reductions in 14 of the 16 predicted values. The second column shows that a bagging of MLPs leads to

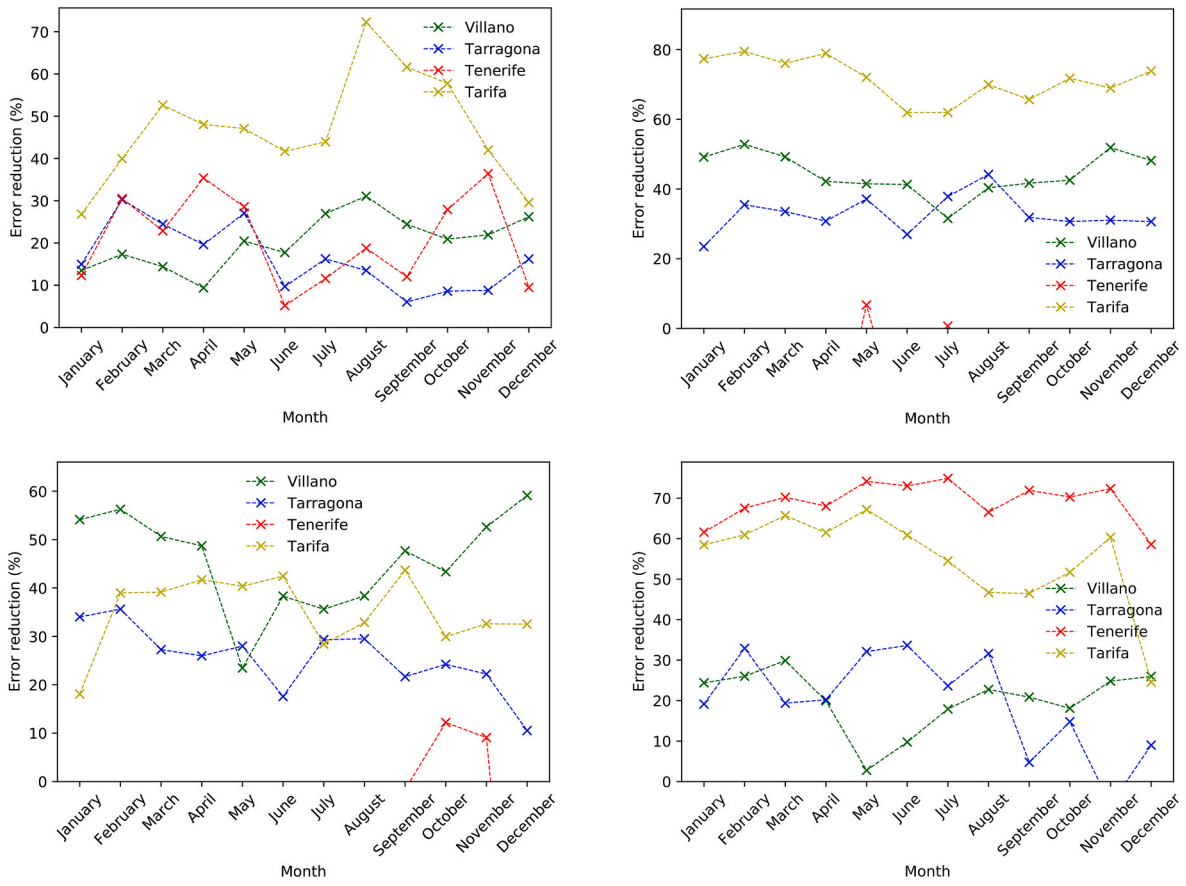


Fig. 12. Error reduction in *Hm0, Tm02, Tp and Dmd* achieved by our ensemble model for each month and location.

further error reduction in all the locations and output variables. GBDTs (LightGBM), which apply bagging and boosting, perform slightly better on average than a bagging of MLPs, although in some cases MLPs provide better results. Hence it is a good idea to combine both approaches. The last column depicts the results of an ensemble of LightGBM and a bagging of MLPs. This solution achieves the highest accuracy for all the locations and variables explored. For instance, compared to LGBM, ensembles further reduce the error ranging from 0.4% (Tarragona, *Tp*) to 13.1% (Tarragona, *DirM*).

Regarding each particular case study, predictions for Villano yields error reductions ranging from 20.0% for *Hm0* to 49.0% for *Tp*. For Tarragona, our solution achieves error reductions from 17.8% for *Hm0* to 32.9% for Tm02. Tarifa is the location where the numerical model is improved by a larger margin: 36.6% for *Tp*, 48.0% for *Hm0*, 58.2% for *DirM*, and 73.8% for *Tm02*. Tenerife is a particular case. While it is observed an error reduction in Hm02 and DirM (21.7% and 69.2%, respectively), the periods *Tm02* and *Tp* do not seem to have relationship with the inputs since none of our models was able to improve the accuracy. Hence, and only in this particular case, the best solution is to use the numerical model without any correction for these two variables.

Figs. 8–11 include scatter plots for the most relevant variable, *Hm0*. Each location includes on the left the baseline scatter plot, i.e., predictions from the numerical model, and, on the right, scatter plots from the predictions of our best ML model are shown for each location. It can be observed how ML models greatly improve accuracy and reduce data dispersion although we can also observe a remarkable underestimation for extreme wave height values, especially in the models developed for Tarragona and Tarifa. This can be explained by the general overestimation of the numerical model in these two locations that forces the ML model to adopt a descending trend.

Table 4 summarizes the accuracy enhancement for each case study. Both mean absolute error and the correlation coefficient are shown as error metrics for each location and output variable.

A seasonality study of the results has been performed to check if the behavior of the ensemble model depends on the type of sea state. This study aims to observe if there is any pattern in the results that could be associated with relative calms (typical situation during the Summer months with low values in wave height and periods) or heavy storms (usually in Winter months, with the highest heights and periods).

In Fig. 12 the results of the study are shown for the four locations and parameters. In the first one (upper left) corresponding to *Hm0*, we can observe that for Villano and Tarifa, the maximum error reduction takes place during Summer, whereas for Tenerife is in Spring and Autumn and for Tarragona is between February and May, We should consider that each location is affected by different wave conditions and this can explain the different trends in the curves. The only similitude that can be appreciated is a descent in June and low values in December and January. For the mean period *Tm02* (upper right) no clear trend is observed, but again, Villano and Tarifa present similar curves with the minimum values around July while Tarragona has the maximum in August. It is a very stable parameter and it does not oscillate too much along the year. For the other two graphs, it is remarkable to highlight a fall in May for Villano and a descending trend along the year for Tarragona.

The only conclusion that can be extracted from this study in relation with the results of the ensemble model is that the low values observed in December and January in *Hm0*, when generally the heaviest storms occur could be related with the underestimation observed in the extremes as shown in the scatter plots (Figs. 8–11).

6. Conclusion

Wave parameter estimation has been traditionally carried out by means of numerical models. In this work several machine learning models have been developed to improve the accuracy of wave parameter estimations made by classical numerical models at four different locations of the Spanish coastline. To this end, they provide a correction to the results of the numerical models. We have focused on four physical wave variables: spectral significant height, mean spectral period, peak period and mean direction of origin.

Different machine learning models have been explored: multilayer perceptron, gradient boosting decision trees and ensemble methods that combine both. These machine learning models have been evaluated, both in terms of accuracy and computational cost (see appendix A).

It has been found that these machine learning algorithms reduce the error of the predictions of the numerical model by 36% on average, demonstrating the potential gains of combining machine learning and numerical models. Error reductions from 19.7% to 73.6% have been achieved in 14 out of 16 case studies.

Our approach increases the accuracy of numerical model predictions in those points where real measures were available. Once the models have been trained, our solution does not demand any instrumentation as it relies solely on the predictions from the numerical model.

Notice that in our approach the predictions of the numerical models are used as inputs of the machine learning models, and machine learning algorithms provide corrections for accuracy improvement. A future and more ambitious work would be trying to completely replace the numerical model instead of correcting it. We would also like to identify the patterns used by the machine learning models in order to better understand why the results are different from one location to another.

CRediT authorship contribution statement

S. Gracia: Conceptualization, Methodology, Software, Writing – original draft. **J. Olivito:** Conceptualization, Methodology, Software, Writing – original draft. **J. Resano:** Conceptualization, Methodology, Writing – original draft. **B. Martin-del-Brio:** Conceptualization, Methodology, Writing – original draft. **M. de Alfonso:** Conceptualization, Methodology, Writing – original draft. **E. Álvarez:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by MINECO/AEI/ERDF (EU) (grants TIN2016-76635-C2-1-R, PID2019-105660RB-C21, TIN2017-88841-R and RTC-2015-3358-5), Aragón Government (T58_20R, T27_17R research groups and LMP16_18), and ERDF 2014–2020 "Construyendo Europa desde Aragón".

We would like to thank everyone that made this interesting project possible: Javier Sanchez at Nologin, Peter Bas at Intel and all people that collaborated from Puertos del Estado. Everybody's feedback and ideas were essential to the positive outcome.

Appendix A. hyperparameters selected and training process

Table 5 details the hyperparameters selected in each case study. We decided to optimize the hyperparameters for each location, since each one is very different from the others. We also explored the possibility of optimizing the hyperparameters for each variable within each location, but in our preliminary experiments we observed that variable-level optimization provided very little further improvement and it was very time-consuming. As it was mentioned in Section 3, we applied a time-series cross-validation strategy to avoid over-fitting.

Table 5
Hyperparameter selection for each location

Location	MLP				LightGBM			
	layers	L2 penalty	estimators	bagging fraction	learning rate	bagging fraction	min data leaf	estimators
Villano	5x20	0.2	20	0.80	0.01	0.80	1,500	2,000
Tarragona		0.2	10					500
Tarifa		0.2	20					2,000
Tenerife		0.5	10					500

We have analyzed the complexity of the training process for each model. To this end, we have trained them in a desktop computer equipped with an Intel i7-2600 CPU and 16 GB of RAM. All our models were coded by using the library Scikit-learn 0.19.1 (Pedregosaet al., 2011). Fig. 13 compares the training time for the four approaches. Results for each approach averages all the locations, and all the variables within each location. The approach based on a single MLP, which is the weaker in terms of accuracy, requires a training time of only 12.4 ± 3.9 s. The approach based on bagging MLP, which yields to more accurate predictions, is one order of magnitude more time-consuming, requiring 189.7 ± 118.7 s on average. It is remarkable the fact that LightGBM models (third approach), which turn out to be slightly more accurate than a bagging of MLPs, are much faster as they only demand 74.8 ± 61.5 s. Finally, results from the ensemble of both bagging of MLPs and LightGBM, are just the sum of both training times (the time required for averaging is negligible).

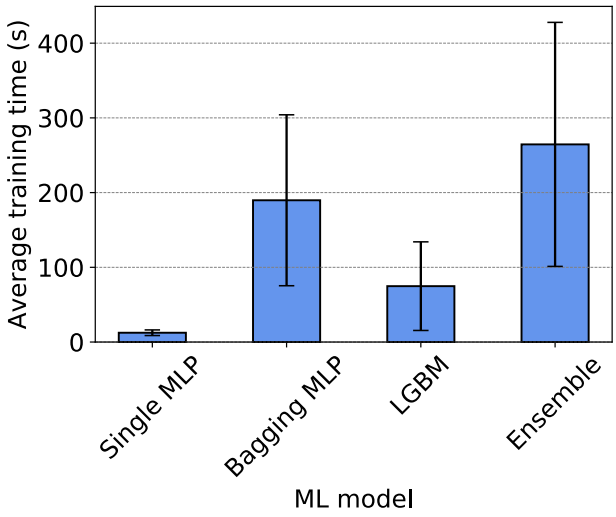


Fig. 13. Average training time of the machine learning models of our study.

References

Alexandre, E., Cuadra, L., Nieto-Borge, J.C., Candil-García, G., del Pino, M., 2018. S. Neurocomputing 275, 818–828.

Bahaj, A.S., 2011. Generating electricity from the oceans. Renew. Sustain. Energy Rev. 15, 3399–3416.

Breiman, L., 1996. Bagging predictors" Mach Learn 24, 123.

Breiman, Leo, 2001. "Random forests", Breiman, L. Machine learning, 45, 5.

Cornejo-Bueno, L., Garrido-Merchán, E.C., Hernández-Lobato, D., Salcedo-Sanz, S., Bayesian optimization of a hybrid system for robust ocean wave features prediction.

Cuadra, L., Salcedo-Sanz, S., Nieto-Borge, J.C., Alexandre, E., Rodríguez, G., 2016. Computational intelligence in wave energy: comprehensive review and case study. Renew. Sustain. Energy Rev. 58, 1223–1246.

Dixit, P., Londhe, S., 2016. Prediction of extreme wave heights using neuro wavelet technique. Appl. Ocean Res. 58, 241–252.

Elleson, et al., April 2020. An application of a machine learning algorithm to determine and describe error patterns within wave model output. Coast Eng. 157, 103595.

Etemad-Shahidi, A., Mahjoobi, J., 2009. "Comparison between M5′ model tree and neural networks for prediction of significant wave height in Lake Superior". Ocean. Eng. 36 (Issues 15–16), 1175–1181.

Falcão, A.F., 2010. Wave energy utilization: a review of the technologies. Renew. Sustain. Energy Rev. 14, 899–918.

Filippo, A., Torres, A.R., Kjerfve, B., Monat, A., 2012. Application of Artificial Neural Network (ANN) to improve forecasting of sea level. Ocean Coast Manag. 55, 101–110.

Géron, A., 2020. Hands-On Machine Learning with Scikit-Learn, second ed. Keras & Tensorflow. O'Reilly.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Guolin, Ke, et al., 2017. LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, vol. 17. NIPS.

Haykin, S., 1999. Neural Networks. A Comprehensive Introduction. Prentice Hall, New Jersey.

James, S.C., Zhang, Y., O'Donncha, F., 2018. A machine learning framework to forecast wave conditions. Coast Eng. 137, 1–10.

Jerome, H., 28 February 2002. Friedman, "Stochastic gradient boosting". Comput. Stat. Data Anal. 38 (Issue 4), 367–378.

Kantardzic, M., Data Mining: Concepts, Models, Methods, and Algorithms Wiley-IEEE Press, 2011. Salcedo-Sanz, "A hybrid genetic algorithm—extreme learning machine approach for accurate significant wave height reconstruction". Ocean Model. 92, 115–123, 2015.

Krishna-kumar, N., Savitha, R., Al-Mamun, A., 2017. Ocean wave height prediction using sequential learning neural networks. Ocean. Eng. 129, 605–612.

LightGBM examples. https://github.com/microsoft/LightGBM/blob/master/examples/README.md#machine-learning-challenge-winning-solutions. (Accessed 18 September 2020).

Liu, L., Wang, D., Peng, Z., 2016. Path following of marine surface vehicles with dynamical uncertainty and time-varying ocean disturbances. Neurocomputing 173, 799–808.

López, I., Andreu, J., Ceballos, S., de Alegría, I.M., Kortabarria, I., 2013. Review of wave energy technologies and the necessary power-equipment. Renew. Sustain. Energy Rev. 27, 413–434.

López, I., López, M., Iglesias, G., 2015. Artificial neural networks applied to port operability assessment. Ocean. Eng. 109, 298–308.

Mahjoobi, J., Mossabeb, E.A., 2009. Prediction of significant wave height using regressive support vector machines. Ocean. Eng. 36, 339–347.

Makarynskyy, O., 2004. Improving wave predictions with artificial neural networks. Ocean. Eng. 31, 709–724.

Makarynskyy, O., 2006. Neural pattern recognition and prediction for wind wave data assimilation. Pac. Oceanogr. 3 (2), 76–85.

Malekmohamadi, I., Ghiassi, R., Yazdanpanah, M.J., 2008. Wave hindcasting by coupling numerical model and artificial neural networks. Ocean. Eng. 35 (Issues 3–4), 417–425.

Marsland, S., 2009. Machine Learning: an Algorithmic Perspective. CRC Press.

Pashova, L., Popova, S., 2011. Daily sea level forecast at tide gauge Burgas, Bulgaria using artificial neural networks. J. Sea Res. 66 (Issue 2), 154–161.

Pedregosa, et al., 2011. Scikit-learn: machine learning in Python. JMLR 12, 2825–2830.

Pooja, J., Deo, M.C., Latha, G., Rajendran, V., 2011. Real time wave forecasting using wind time history and numerical model. Ocean Model. 36 (Issues 1–2), 26–39.

Puscasu, R.M., 2014. Integration of artificial neural networks into operational ocean wave prediction models for fast and accurate emulation of exact nonlinear interactions. Procedia Computer Science 29, 1156–1170.

SWAN Scientific and Technical Documentation, 2009. Delft University of Technology.

Timofeev, R., 2004. Classification and Regression Trees (CART) Theory and Applications. Master's Thesis. Humboldt University Berlin.

Vanem, E., 2011. Long-term time-dependent stochastic modelling of extreme waves. Stoch. Environ. Res. Risk Assess. 25 (2), 185–209.

Yasseri, S.F., Bahai, H., Bazargan, H., Aminzadeh, A., 2010. Prediction of safe sea-state using finite element method and artificial neural networks. Ocean. Eng. 37 (Issues 2–3), 200–207.

Yin, J., Zou, Z., Xu, F., 2013. Sequential learning radial basis function network for real-time tidal level predictions. Ocean. Eng. 57, 49–55.

Zhang, Z., Li, C.W., Qi, Y., Li, Y.S., 2006. Incorporation of artificial neural networks and data assimilation techniques into a third-generation wind-wave model forwave forecasting. J. Hydroinf. 8 (1), 65–76.

Zheng, Z., Sun, L., 2016. Path following control for marine surface vessel with uncertainties and input saturation. Neurocomputing 177, 158–167.