Journal of
Marine Science and Engineering

MDPI

Article

Gradient Boosted Trees and Denoising Autoencoder to Correct Numerical Wave Forecasts

Ivan Yanchin and C. Guedes Soares *

Centre for Marine Technology and Ocean Engineering (CENTEC), Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal; ivan.yanchin@tecnico.ulisboa.pt
* Correspondence: c.guedes.soares@centec.tecnico.ulisboa.pt

Abstract: This paper is dedicated to correcting the WAM/ICON numerical wave model predictions by reducing the residue between the model's predictions and the actual buoy observations. The two parameters used in this paper are significant wave height and wind speed. The paper proposes two machine learning models to solve this task. Both models are multioutput models and correct the significant wave height and wind speed simultaneously. The first machine learning model is based on gradient boosted trees, which is trained to predict the residue between the model's forecasts and the actual buoy observations using the other parameters predicted by the numerical model as inputs. This paper demonstrates that this model can significantly reduce errors for all used geographical locations. This paper also uses SHapley Additive exPlanation values to investigate the influence that the numerically predicted wave parameters have when the machine learning model predicts the residue. To design the second model, it is assumed that the residue can be modelled as noise added to the actual values. Therefore, this paper proposes to use the denoising autoencoder to remove this noise from the numerical model's prediction. The results demonstrate that denoising autoencoders can remove the noise for the wind speed parameter, but their performance is poor for the significant wave height. This paper provides some explanations as to why this may happen.

Keywords: significant wave height; wind speed; denoising autoencoders; autoencoders; gradient boosting; machine learning

check for updates

Citation: Yanchin, I.; Guedes Soares, C. Gradient Boosted Trees and Denoising Autoencoder to Correct Numerical Wave Forecasts. *J. Mar. Sci. Eng.* **2024**, *12*, 1573. https://doi.org/10.3390/jmse12091573

Academic Editors: Nawin Raj, Lila Singh-Peterson and Nathan Downs

Received: 27 July 2024
Revised: 29 August 2024
Accepted: 3 September 2024
Published: 6 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Wind wave parameter forecasting is known to be a difficult task. Wind wave generation is a non-stationary stochastic process, with its parameters changing with time. Several numerical models have been developed to forecast wind wave parameters such as WAVEWATCH III [1], WAM [2] and SWAN [3]. These models try to capture the physics behind the wave generation process and compute the desired wave parameters using the supplied input data, and they represent the present state-of-the-art [4].

These models have been widely used to produce hindcast databases for 30 to 40 years. Comparisons of the databases show that, on average, these models perform similarly in the range of up to moderate wave heights but diverge for high waves [5], which are precisely the ones that are more critical to maritime operations. Weather ship routing is essential for planning maritime operations. These systems have been used for a long time, and their usefulness depends to a large extent on the quality of weather forecasts [6]. They include models that represent the ship's behaviour under the effect of waves and optimisation models that identify the optimal routes for different objectives, such as minimum time, minimum fuel consumption and maximum operational safety [7].

Research shows that considering weather conditions when planning a ship route can reduce the voyage duration by 7% for short routes [8]. Specifically, taking wind waves into account and avoiding areas with rough wave conditions can help reduce the associated costs by up to 18% [9].

Unresolved scientific questions still affect the numerical model's design and the accuracy of the model's output [10]. Tolman [10] points out that proper numerical modelling of wave propagation using a discrete spectral model is a difficult task, specifically due to the need to accurately model the advection of swell energy. These difficulties become even more serious in shallow waters because the nonlinear propagation effects become important. Another issue is that the understanding of wave physics is less developed than of the wave propagation and kinematic processes [10]. Moreover, numerical models require significant resources to compute the result, which may make it infeasible to use such models for real-time forecasts [11–13]. The exact computations of nonlinear wave interactions can be prohibitively expensive [10]. Another important issue is that forecast accuracy for long-term forecasts degrades compared to short-term forecasts, while long-term forecasts may be more important in practice, as they allow the better planning of maritime operations [14].

Machine learning models are known to be able to capture inner patterns of the data without the need to fully understand the processes that generated it [15,16]. Therefore, several attempts were made to overcome the aforementioned accuracy issues using machine learning.

The first group of methods tries to replace numerical models entirely and uses machine learning to compute the desired wave parameters.

Elbisy et al. [17] use several machine learning models of different types to predict significant wave height. Specifically, the paper uses a gradient boosting machine over decision trees (GBT, but the authors use a synonymous term MART for this technique) to solve the same task. Gradient boosted trees outperformed all of the machine learning models, and the neural network that uses radial basis functions has demonstrated the second-best accuracy.

Hu et al. [18] also used GBT, specifically XGBoost. The model takes wind speed and the direction from two buoys as inputs and predicts the wave height and the peak wave period for a third location. The authors compared the accuracy of the GBT model with long short-term models (LSTM), which were trained to predict the same targets, accumulating historical wind speed and wind direction data. The paper shows that GBT outperforms LSTM in terms of accuracy.

Ahn et al. [19] suggest the use of LSTM to predict significant wave height using wind parameters and several previously known significant wave height values. The authors discovered that wind parameters do not significantly affect the prediction accuracy. They speculate this is because the previously known significant wave heights already contain almost all of the information required for the forecast, and wind parameter inclusion does not bring any new information.

Generally, one of the most frequently mentioned advantages of machine learning models over numerical ones is that they can compute the result much faster, sometimes even by orders of magnitude, while providing similar accuracy [11–13,20].

Several attempts were made to improve the numerical models' predictions using deep learning or to enhance the numerical models themselves.

For example, Puscasu [21] attempted to integrate a neural network model into WAVEWATCH III, replacing its numerical non-linear term calculations. Browne et al. [22] used a neural network to adjust the forecasts of WAVEWATCH III for a near-shore area distant from the original forecast area. Campos et al. [14] used a neural network to improve wave parameter forecast by using stacking, a machine learning technique that trains a new model to weigh and average the outputs of multiple other models. Londhe et al. [23] demonstrated that the residue between the actual wave parameters and the forecast can be modelled as a time series and thus predicted for 24 h in advance.

Fan et al. [24] describe a hybrid SWAN-LSTM model to forecast significant wave height. Here, SWAN is used to estimate wind parameters and wave height, while the LSTM-based model is used to refine the SWAN's prediction.

Pirhooshyaran and Snyder [25] predicted significant wave height and spectral wave density. For the significant wave height, the authors used an LSTM-based neural network. For the spectral wave density, they used a sequence-to-sequence model that used the encoder–decoder architecture, with both encoder and decoder being LSTM-based. The authors conclude that these models can successfully predict the required wave parameters. The authors argue that sequence-to-sequence models demonstrate better results than the other studied models. They also point out that multi-layer models are not superior to single-layered ones. Moreover, the authors demonstrate that these model architectures can be used to estimate wave parameters for an unstudied area using the measurements of nearby wave buoys.

Costa et al. [26] use machine learning models to predict the residue between the computed wave parameters and the actual parameter values using the historical hindcast data, both for a particular moment and a recurrent LSTM-based model, as a time series.

As one can see, LSTM is the most commonly used model architecture to predict wind and wave parameters. However, several papers propose using GBT [17,18]. Other papers, such as [25], emphasise that capturing the sequential nature of wave parameters may be necessary for proper prediction.

This paper proposes machine learning models to correct a numerical model's prediction, making long-term predictions more accurate. This paper follows the path suggested in [23,24,26] designing machine learning models to predict the residue between the buoy data and the output of a numerical model that was computed for the locations of the buoys. It is, therefore, possible to say that the task is to correct the output of the numerical model so that it is as close to the actual observations as possible. The parameters of interest are the significant wave height and the wind speed at 10 m height. The WAM/ICON numerical model is used in this paper. This model is the well-known WAM wave model forced by winds from the ICON (Icosahedral Nonhydrostatic) global weather model, developed by the German national weather service (Deutscher Wetterdienst) [27,28].

The rest of this paper is organised as follows. Section 2 covers the results of the exploratory data analysis that investigates the properties of the dataset and the wave and wind generation processes under study. This section also defines the baseline model. Section 3 presents a model that can predict the residue between the buoy observations and the numerical model predictions for a time point using gradient boosted trees. Section 4 suggests representing this residue as noise and describes a sequence-to-sequence encoder–decoder neural network trained to remove this noise. Section 5 compares the models and states this paper's novelty and possible future research directions. Section 6 concludes the paper by summarising the findings.

2. Exploratory Data Analysis

The data used in this paper comes from two sources. The first one is the prediction made using the WAM/ICON model [27,28], and the second data source is the real measurements performed by the buoys near the Iberian Peninsula's shores. The task is to predict the residue between the WAM/ICON model output and the buoy data. Numerical model outputs and the actual measurements cover the period from the 1 January 2022 to the 31 December 2022.

This study uses two buoys. The first buoy has the WMO code 6200082 and is located near the Punta da Estaça de Bares, which is the northernmost point of Spain. This buoy is located at the point (44°07′12.0″ N; 7°40′48.0″ W). The second buoy has the WMO code 6200200 and is located near Faro, a city in southern Portugal. This buoy is located at the point (36°23′24.0″ N; 8°04′12.0″ W). Herein the buoys are called "the Bares buoy" and "the Faro buoy", respectively. Figure 1 shows the locations of the buoys.

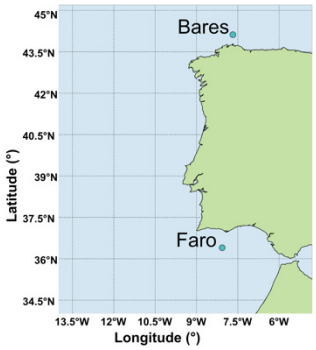

Figure 1. Location of the buoys around the Iberian Peninsula.

The buoy data contains hourly measurements of several parameters of the wind and wave climate; however, this paper only uses the significant wave height (SWH) and the wind speed at 10 m (WS). The buoy data has some missing measurements caused by buoy outages. These gaps need to be filled because missing values may reduce the effectiveness of machine learning models. This study replaces the missing values according to the following rules. If it is an isolated missing measurement (i.e., the measurements exactly before and precisely after the gap exist), its missing value is linearly interpolated. If multiple subsequent measurements are missing, but there still are existing values before and after the gap, the gap is filled with the average value of the parameter computed using the existing parameter values. If the missing measurements are located at the beginning or the end of the dataset (i.e., there were either no existing measurements before or no existing values after the gap), the respective time steps are removed from the dataset. These rules are applied to each buoy and each parameter independently.

The numerical model output contains multiple parameters that describe different wind and wave climate properties. Table 1 lists these parameters. A detailed description of the wave parameters produced by the WAM/ICON model can be found in [29]. WAM/ICON model data contains wave parameter predictions with 3 h steps. These two datasets are synchronised to properly compare the buoy data and the numerical model's predictions. A corresponding data buoy measurement is found for each numerical model data point. The buoy measurements for which no corresponding numerical model data points exist are removed from the dataset.

Table 1. The parameters produced by the numerical model.

Parameter	Description
SWH	Significant wave height (total spectrum).
WS	Wind speed at 10 m (total spectrum).
MWD	Mean wave direction (total spectrum).
TM10	"Energy" wave period (total spectrum).
MDTS	Mean wave direction (swell partition).
PPTS	Peak wave period (swell partition).
MPTS	Mean wave period (swell partition).
SHTS	Significant wave height (swell partition).
MDWW	Mean wave direction (wind sea partition).
MPWW	Mean wave period (wind sea partition).
PPWW	Peak wave period (wind sea partition).
SHWW	Significant wave height (wind sea partition).

After filling the missing values of the buoy data and dataset synchronisation, the dataset contains 2920 data points for the Bares buoy and 1662 data points for the Faro buoy. The dataset is divided into training and testing subsets in a 80%:20% proportion, which is commonly used in machine learning research, therefore producing 2336 training and 584 testing data points for the Bares buoy and 1078 training and 584 testing data points

for the Faro buoy. The test dataset contains precisely the same number of data points for both buoys to make the models' performance comparable, each of which is trained for a particular buoy. For the same reason, the training dataset uses the same time points for both buoys. This results into a smaller training dataset for the Faro buoy. It is, however, more important to have the test sets of equal size for different buoys, to make it possible to compare the accuracy of the models between the buoys. Nevertheless, one should expect less model accuracy for the Faro buoy due to the smaller size of the training dataset.

The dataset also contains the residues for the SWH and the WS, which are computed as the difference between the values measured by the buoys and those calculated by the numerical model.

$$\mathrm{R}_p = \mathrm{B}_p - \mathrm{M}_p \tag{1}$$

$$C_p = M_p + \hat{R_p} \tag{2}$$

Equation (1) shows how the residues are computed, and Equation (2) shows how the numerical model output is corrected using the prediction of an ML model. Here R_p is the residue between the buoy data and the numerical model output, B_p is the buoy data, M_p is the model output, C_p is the corrected model output, $\hat{R_p}$ is the residue predicted by an ML model, and subscript p indicates the particular parameter, either the SWH or the WS.

The data from the buoys and the model have been combined into a single dataset. To distinguish the features produced by the numerical model, they are named with the "*model_*" prefix, and the values that were measured by the buoys do not have a prefix. Each feature has the suffix "*_bares*" if computed or measured for the Bares buoy and "*_faro*" if it relates to the Faro buoy.

The SWH and the WS change over time and, therefore, can be represented as time series, i.e., an ordered set of data points associated with a time point. One of the most common approaches to time series forecasting is using autoregressive models. These are the models that use historical data to predict future changes in the target features over time. However, for this approach to succeed, the future values of the target feature must be correlated with their past values. Autocorrelation plots are used to evaluate this. Autocorrelation plots show the correlation between a feature value and its lags (i.e., the historical values of the feature). The height of the peaks on the plots shows the magnitude of the correlation coefficient, and the semi-transparent area indicates the confidence interval. If a peak goes beyond the confidence interval, then this autocorrelation value is likely caused by the real correlation instead of statistical fluctuation. Partial autocorrelation plots are organised similarly but show the autocorrelation between the value and its lags without the influence of the intermediate lags. Speaking in terms of ARIMA models, partial autocorrelation defines the parameter of the $AR(*)$ term, and autocorrelation defines the parameter of the $MA(*)$ term.

Figure 2a shows the autocorrelation and partial autocorrelation plots for the SWH residue computed for the Bares buoy. As one can see, the assumption that the SWH depends on the previous values is correct. For the Bares buoy, the autocorrelation stays significant for several lags. Only starting from the 23rd lag, the peaks are mostly below the confidence threshold. Regarding partial autocorrelation, the peaks go below the confidence threshold sooner, indicating that the significant correlation observed for autocorrelation is mostly achieved due to the cumulative effects. Both in autocorrelation and partial autocorrelation the same pattern can be observed. There is a global decrease trend; however, sometimes the peaks increase in value. The periodical nature of wind waves can explain such increase and decrease periods. Also, this effect can be explained by daily seasonality. One can see that this pattern repeats approximately every four peaks. Considering that the time difference between a pair of subsequent peaks is 3 h, one can say that there is a 12-h global period in autocorrelation. This period corresponds to the known tide–ebb period.

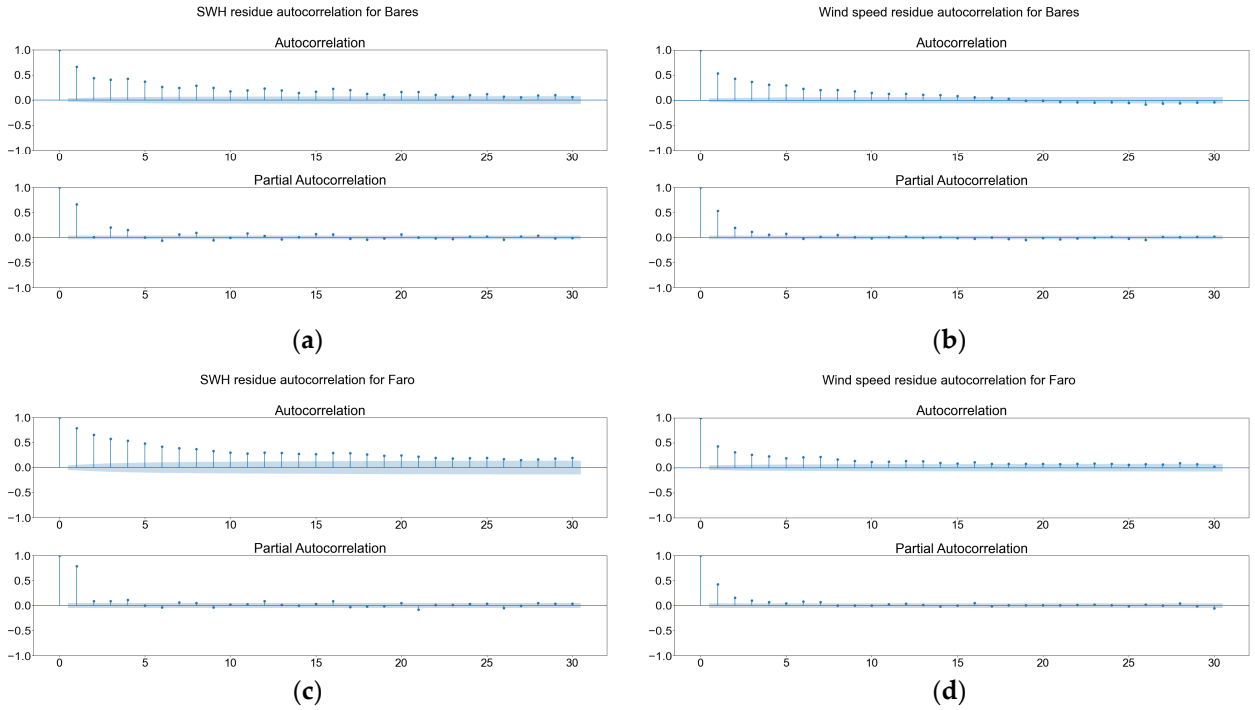

Figure 2. Comparison between the buoy data, the numerical model prediction and the corrected numerical model prediction. (**a**) Autocorrelation and partial autocorrelation plots for the SWH of the Bares buoy. (**b**) Autocorrelation and partial autocorrelation plots for WS of the Bares buoy. (**c**) Autocorrelation and partial autocorrelation plots for the SWH of the Faro buoy. (**d**) Autocorrelation and partial autocorrelation plots for the WS of the Faro buoy.

Figure 2b shows the autocorrelation and partial autocorrelation plots for the WS residue computed for the Bares buoy. One can see that autocorrelation decreases steadily, without the decrease–increase pattern that was observed for the SWH. This means that wind speed does not have significant daily seasonality.

Figure 2c shows the autocorrelation and partial autocorrelation plots for the SWH residue but computed for the Faro buoy. In this figure, one can see the same pattern as the Bares buoy. Autocorrelation stays significant for quite a long time, but partial autocorrelation drops significantly after the first lead. However, there are still some peaks above the confidence threshold after the first peak below the confidence threshold. One can also see the same periodical pattern that is due to seasonality. In the case of wind speed near the Faro buoy, as seen in Figure 2d, one can observe the same seasonality indicated by the decrease–increase pattern with the period of approximately 6 peaks or 18 h.

Baseline Model

A baseline model is a model that is used as a measurement standard. If a model under study outperforms the baseline, then this model is considered to succeed in solving the task. In the case of the current study, which is dedicated to correcting the errors of a numerical model, the baseline is the numerical model itself. Therefore, the baseline metrics are the metrics that are computed for the numerical model outputs compared to the buoy data. This study uses mean squared error (MSE), mean absolute error (MAE) and normalised root mean squared error (NRMSE) as metrics. The baseline metric values, computed for the numerical model without any correction, are presented in Table 2.

Table 2. The baseline metrics.

Metric	Bares Buoy		Faro Buoy	
	SWH	WS	SWH	WS
MSE	0.2132	5.2553	0.2062	2.5962
MAE	0.3528	1.8607	0.3332	1.1852
NRMSE	0.1235	0.2749	0.2117	0.2590

As can be seen from the baseline metrics, the WAM/ICON model has relatively low errors when predicting both the SWH and the WS, with the errors for the SWH being smaller than for the WS. For the Faro buoy, the metrics are smaller than for the Bares buoy. But, the pattern of the SWH metrics being much smaller than the WS metrics holds for the Faro buoy.

3. Predicting the Residue for a Time Point

This section is dedicated to predicting the residue between the WAM/ICON model output and the actual buoy data for a single time instant using only the data of that time instant. This means the predictor model can only use the wave parameters that are computed for this time instant, i.e., the predictor cannot use historical data. This approach assumes that by using the patterns of other wave parameters produced by the numerical model, an ML model can infer the error of the target wave parameters.

One of the important ideas related to predictive models is the lead time. The lead time is the number of time steps in the future for which a model is required to predict the target value. Since the model predicts the newer values using historical values, starting at some point, it will use the previously predicted values to obtain the new ones. This is one of the reasons why the prediction error increases when the lead time increases: prediction errors accumulate. However, the approach described in this section assumes the prediction of the target value for a single time step, using only the wave parameters computed for this step. As a result, the models described in this section do not depend on the previous predictions when predicting the target value for a particular lead time. Therefore, one may expect that the prediction error of the ML models will be consistent regardless of the lead time.

3.1. The Predictor Model

Decision trees have already demonstrated good results when predicting wind wave parameters [17,18]. However, to the authors' knowledge, this type of model has not been used to correct the prediction of a numerical model. Gradient boosting machine over decision trees (GBT) is an ensemble machine learning technique that assumes the creation of many simple decision trees. In contrast with Random Forest models [30], only the very first tree is trained to predict the target value, while all of the subsequent trees are trained to predict the error of the ensemble model constructed already. The theory behind GBT is based on the observation that a set of weak estimators together can make up a single strong estimator, which was first demonstrated for the field of cryptography [31].

The difference between the GBT model presented in this paper from the one discussed in [18] is that the former uses the values produced by a numerical model to estimate its error and then correct its prediction, while the latter is trained to infer the patterns that connect the wind parameters to the wave parameters.

The task of the model being designed in this section is to predict both the SWH and the WS together as multiple outputs of the same model. These two parameters are closely related to each other; therefore, the authors argue that a single model can predict them together, possibly demonstrating better results than two models each predicting a single parameter. It is also expected that a single model's knowledge of the combination of patterns for different weather parameters (SWH and WS specifically) can, at least partially, compensate for the Faro buoy's smaller training dataset. This is a multioutput regression task. There are multiple implementations of GBT, and CatBoost [32], XGBoost [33] and

LightGBM [34] are the most well-known among them. However, at the time of writing, only CatBoost is known to have a production-ready implementation of a multioutput GBT regressor. Therefore, this paper uses CatBoost. During this research, two GBT models have been created: one to predict both the SWH and the WS residues for the Bares buoy and the second to do the same for the Faro buoy.

One advantage of GBT models over the other types of ML models is that such models are based on the comparison of numbers instead of arithmetical operations using them and that they internally use an entropy-based feature selection mechanism. Therefore, there is no need to perform feature scaling, as the scale of values does not affect their comparison. Moreover, there is no need to perform feature selection, as during the training process GBT models select which feature to use to split a particular tree using the information gain criteria [32,35].

CatBoost provides multiple hyperparameters that can be tuned to improve the model's prediction accuracy. Table 3 shows the hyperparameters tuned during the current research to improve the model performance. Both models used the same set of hyperparameters of interest and the same hyperparameter ranges. The ranges for the hyperparameters were selected according to the recommendations of the official documentation of CatBoost. In both cases, random search was used to select the best hyperparameter set, and MSE was chosen as the minimisation target of the search process. Random search was used because its implementation provided by the scikit-learn Python package allows for continuous parameter domains specified by distributions, and some of the hyperparameters are of this kind. The search was stopped after 10 iterations, because this was found to produce good results within a reasonable time. The search process was run for each of the buoys independently.

Table 3. GBT hyperparameter ranges.

	Domain	Type	Description
depth	{6, 7, 9, 11}	Discrete	The maximum depth of individual decision trees.
n_estimators	{300, 500, 1000}		The number of individual decision trees to create.
l2_leaf_reg	[0; 25]	Uniformly distributed	L2 regularization factor.
learning_rate	[0.001; 0.9]		Regulates the magnitude gradient step of the algorithm and can be adjusted to prevent overfitting.

Table 4 shows the hyperparameter values selected by the random search. As one can see, for both buoys, the hyperparameter search selected 7-level deep trees. The other parameters are different. For the Faro buoy, the search yielded greater values for all observed hyperparameters. This may be explained by the fact that for the Faro buoy, the dataset has a smaller number of samples, therefore requiring more estimators and a greater learning rate to be able to generalise to the data and a greater L2 regularisation coefficient to compensate for a more complex model and to prevent its overfitting.

Table 4. Hyperparameters selected for the models.

	Bares	Faro
depth	7	7
n_estimators	300	1000
l2_leaf_reg	3.0423	5.3705
learning_rate	0.6746	0.7165

3.2. Investigating the Models' Accuracy

Table 5 shows the metric values computed for the test dataset using the GBT models trained with the discovered hyperparameters. These metrics are computed by applying

the models to the test inputs and then applying the result as a correction to the numerical model outputs, as shown by Equation (2). The corrected numerical model's output is then compared to the buoy data.

Table 5. Metrics computed for the test dataset using the GBT models.

Metric	Bares Buoy		Faro Buoy	
	SWH	WS	SWH	WS
MSE	0.1968	2.6523	0.1262	2.3633
MAE	0.3413	1.2254	0.2537	1.1774
NRMSE	0.1187	0.1953	0.1656	0.2471

Comparing the metric values in Table 5 with the baseline metrics shown in Table 2, one can see that all ML models managed to improve the prediction accuracy according to all of the metrics. For the Bares buoy, the improvement of the SWH is not very significant. However, the WS has significantly improved for this buoy, the MSE has changed from 5.26 to 2.65, and the other metrics are also lower than the baseline. For the Faro buoy, SWH accuracy changes more significantly. For example, the MSE changes from 0.2062 to 0.1262. Compared to the Bares buoy, the improvement of WS for the Faro buoy is not that significant, but the metrics are still notably lower.

Figure 3 shows the comparison between the actual buoy data, the numerical model prediction and the corrected numerical model prediction. As one can see, the corrected numerical model prediction is almost always much closer to the buoy data than the non-corrected model. The horizontal segments of the Faro buoy plots indicate the missing data replaced with the mean value. Because this data imputation is performed in the exploratory data analyses step, here it is used "as is", and therefore is used to compute the metric values. However, it is not expected to significantly affect the findings of this paper, as all models (including the baseline) are tested using the same data. As one can see from Figure 3, the GBT models usually fail to properly predict the target near the peak values. This is the expected behaviour of decision tree algorithms. Such algorithms are expected to smoothen peak values, as they are based on averaging values on the leaves of the trees.

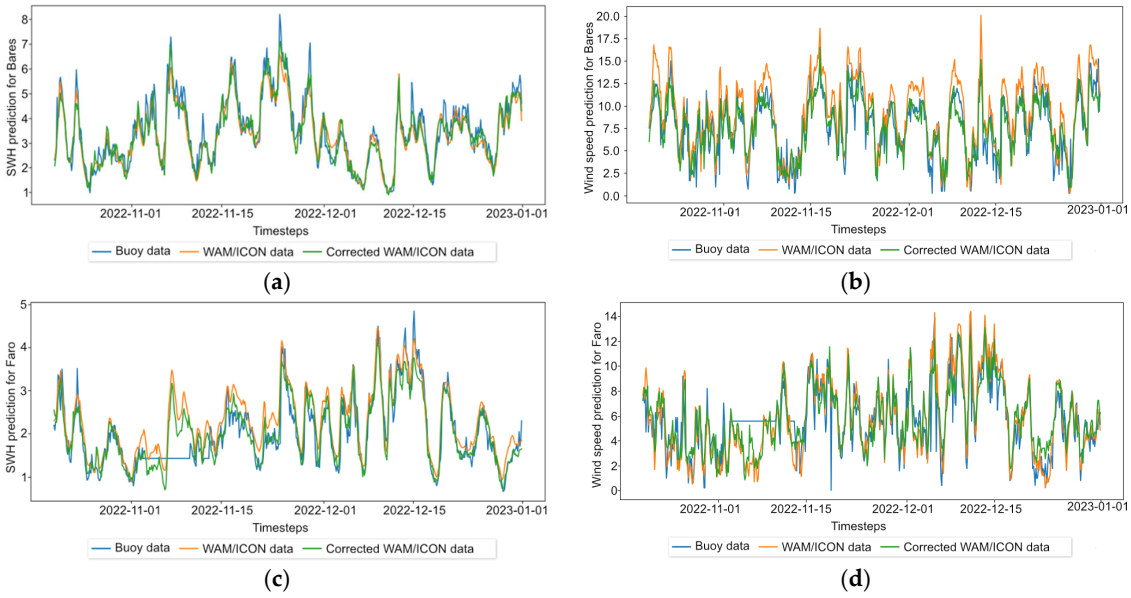

Figure 3. Comparison between the buoy data, the numerical model prediction and the numerical model prediction corrected with GBT. (**a**) Corrected WAM/ICON data for the SWH for the Bares buoy. (**b**) Corrected WAM/ICON data for the WS for the Bares buoy. (**c**) Corrected WAM/ICON data for the SWH for the Faro buoy. (**d**) Corrected WAM/ICON data for the WS for the Faro buoy.

Although the metrics for the corrected SWH are almost the same as for the baseline, they nevertheless represent an improvement in numerical model correction. Because the numerical model predictions are very accurate on their own, it is difficult to improve these predictions significantly. Figure 4 shows the comparison between the buoy data, numerical model output and the corrected numerical model output for the SWH parameter for the Bares buoy for a smaller subrange of the test dataset. As one can see, the corrected numerical model prediction is usually closer to the buoy data than the non-corrected prediction.

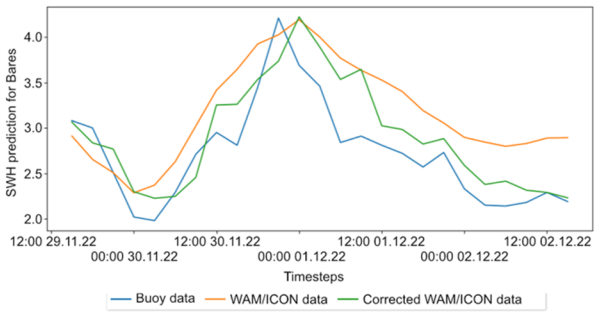

Figure 4. Comparison between the buoy data, the numerical model prediction and the corrected numerical model prediction for a subrange of the training dataset.

3.3. Investigating the Importance of the Features

Explaining an ML model's decisions is known to be a tricky task. However, there exist some techniques that can be used for this. SHapley Additive exPlanation (SHAP) values are one such technique [36,37]. SHAP values represent the effect a particular feature has on the ML model's output. SHAP values are computed for each object separately but can be combined to get the overall picture. On the plots, the features are sorted from the most important (on the top of the plot) down to the least important. The horizontal axis of the plots denotes the effect a particular feature has on the prediction. The sign of a SHAP value indicates whether a feature increases or reduces the predicted value, and the magnitude of a SHAP value shows the magnitude of this effect. The colour of the point associated with a SHAP value denotes the magnitude of the feature itself. As a result, using the SHAP value plot, one can determine what effect a feature has on the target value and how this effect changes when the value of the feature changes.

Figure 5 shows the SHAP values for the GBT models. As one can see from the plot, when predicting the SWH residue for the Bares buoy (Figure 5a) the most important feature is the wind speed while all of the other features have significantly smaller effects. Interestingly, the numerically modelled SWH is not very important when correcting the SWH. At the same time, the SWH is not very important when predicting the WS residue for the Bares buoy (Figure 5b). The most important feature in this case is the SHWW. Similar to the SWH, when correcting the WS, the numerically modelled WS value is not very important.

In contrast to the Bares model, the Faro model assigns more importance to the MPTS when correcting the SWH (Figure 5c). The Faro model also considers all of the features almost equally important when correcting the SWH. When correcting the WS, the Faro model considers the numerically modelled WS value to be the most important feature and the PPTS to be the second most important (Figure 5d), while the Bares model considers the SHWW to be the most important and the predicted WS value to be the seventh most important. Interestingly, the Faro model considers the SWH to be more important when predicting the WS (the third most important parameter), in contrast with the Bares model (the seventh most important parameter). Note that for both of the buoys, a single multi-output model is used to predict both the SWH and the WS. Therefore, the SHAP values show that the same model gives different importance to the same features when predicting different target values.

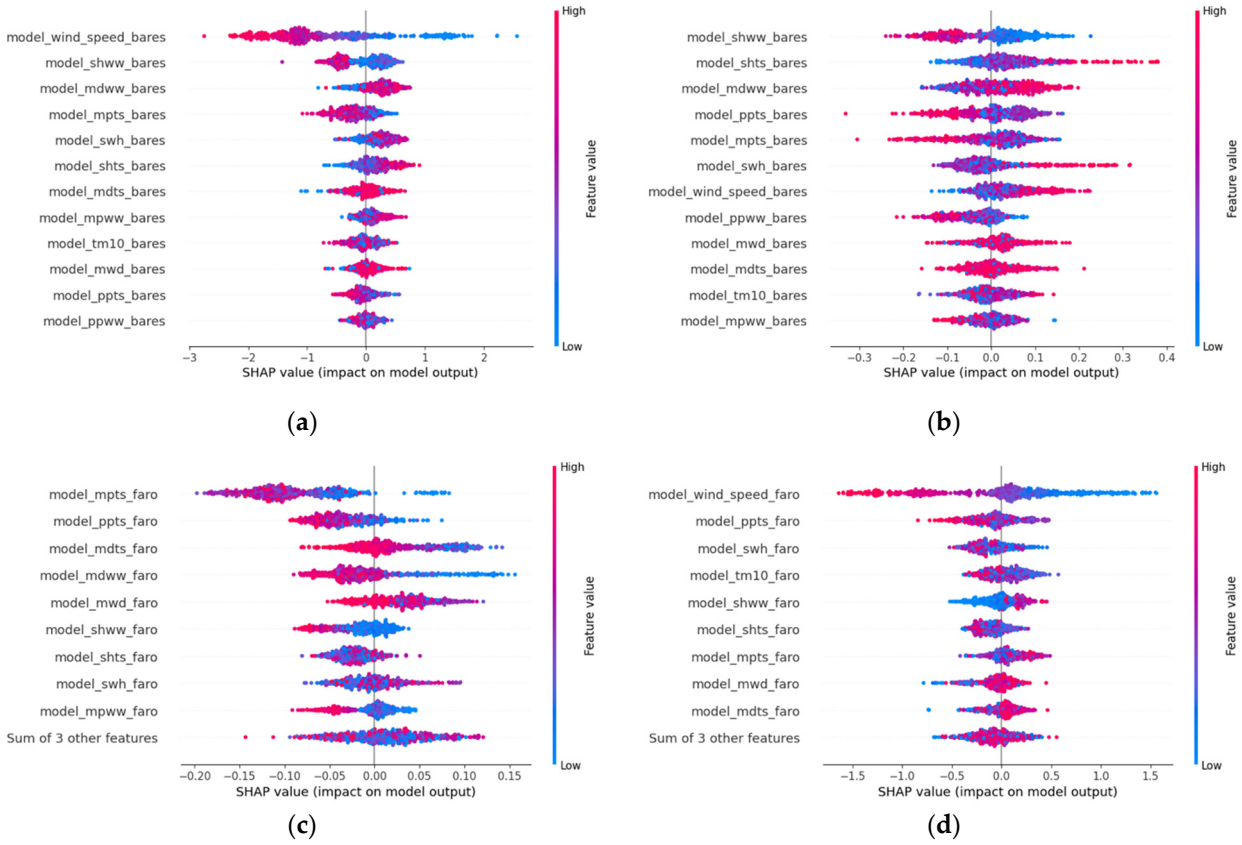

Figure 5. SHAP values for the GBT models. (**a**) SHAP values for the SWH for the Bares buoy. (**b**) SHAP values for the WS for the Bares buoy. (**c**) SHAP values for the SWH for the Faro buoy. (**d**) SHAP values for the WS for the Faro buoy.

One can notice that when predicting the SWH for the Bares buoy the most important parameter is the original WS, but the improvement in the parameter is not very significant. Similarly, when correcting the WS prediction for the Faro buoy, the original WS parameter is also considered the most important and the improvement is not significant. Based on this, one can conclude that the WS is not a good predictor for the target values regardless of their nature. However, the following observations conflict with this assumption:

- The gradient boosting (and other tree-based algorithms) uses the information gain criterion to select the parameters to split the trees on [32,35]. The fact that the WS parameter is considered the most important (using the SHAP values) means that splitting on it brings the most significant information gain. Therefore, this parameter indeed significantly improves the prediction of the target parameters, otherwise it would not be selected to split the tree.
- When predicting the SWH parameter residue for the Faro buoy, the WS parameter is not very important (according to the SHAP values), but the forecast improvement is as significant as it is for the WS parameter. This observation conflicts with the initial assumption that the presence of the original WS parameter reduces the accuracy of the model because the lack of this parameter does not result in a significant improvement.

The training dataset has different amounts of data for different buoys. The amount of information gained when splitting a tree at some parameter may be affected by the size of the dataset available for this parameter, as a larger dataset may more thoroughly represent the parameter domain therefore affecting its importance and the associated SHAP value. Because of the different dataset sizes, it may be incorrect to compare the importance of the same parameter but for the different buoys. Investigating this may be a subject of the future research.

4. Sequence-to-Sequence Prediction

The models described in the previous section can predict the residue between the buoy data and the output of a numerical model using the other parameters produced by this model. However, additional parameters are not always available. This section proposes models that are trained with only the SWH and the WS data. Similarly to the previous section, these models are designed to correct the SWH and the WS together as multiple outputs of a single model. Similarly to the GBT, one may expect that training a single model for both parameters will compensate for the smaller training dataset of the Faro buoy.

4.1. Denoising Autoencoders

Let B_p be a value for a parameter p (either SWH or WS) observed by a buoy and M_p be the same parameter p but computed by a numerical model for the same geographical location and time. Then $R_p = B_p - M_p$ is the residue between the model and buoy data. Wind and wave parameters are known to be stochastic processes, therefore R_p can be considered stochastic noise drawn from an unknown distribution. Therefore, the task of correcting the prediction of a numerical model can be represented as a task of denoising a signal. Neural networks demonstrated good results when trained to capture the properties of unknown distributions, and denoising autoencoders are a special family of neural networks designed specifically for the denoising task [38].

An autoencoder consists of two parts: an encoder and a decoder. An encoder is a neural network trained to approximate a function $E(x) = h$ where $x \in \mathbb{X}$, $h \in \mathbb{H}$, where $\mathbb{X} \subset \mathbb{R}^n$, $\mathbb{H} \subset \mathbb{R}^m$ and $m \ll n$. Then, a decoder is trained to recover x from h, i.e., it is trained to approximate a function $D(h) = \hat{x}$ [15]. Usually, the encoder and decoder are combined into a single neural network that is trained to accept the input data, compress it to a smaller space and then recover the original value. In such cases, the loss function used to train such networks is $L(x,\ D(E(x)))$. Since the cardinality of h is smaller than the cardinality of x, some of the information in x is lost. However, because the loss function requires the decoder's output to be as close to the encoder's input as possible, the encoder must infer such a mapping between $\mathbb{X}$ and $\mathbb{H}$ that keeps the most important information in x [15]. Vector h computed for a specific object x is often called a representation of x or an embedding of x [15]. Usually, the encoder and decoder are symmetrical, but this is not a requirement [15].

Autoencoders are trained in an unsupervised manner because they are required to produce the same value as the input. In practice, after an autoencoder is trained, its decoder part is removed, and the embeddings produced by the encoder are used as inputs to other models [15]. Usually, autoencoders are explicitly trained to be later used to produce embeddings for the input objects [15].

Denoising autoencoders (DAEs) are a special kind of autoencoders. DAEs are trained not to produce the output that is the same as the input, but to remove the noise from the input data and produce the denoised data [15]. Therefore, DAEs are trained in a supervised manner, as both noisy and clear data are required. The loss function for DAEs is $L(x,\ D(E(x + \epsilon_x)))$ where ϵ_x is the noise term for the object x. Although, in the case of DAEs, both encoder and decoder are usually used in practice (because the decoder produces the denoised data), they can still be used to obtain embeddings for objects.

4.2. Denoising Autoencoder to Correct Numerical Model Prediction

This paper proposes a denoising autoencoder to correct the output of a numerical model so that the result gets closer to the buoy observations. For this, the residue between the buoy data and the numerical model output is considered as noise associated with the numerically modelled data. Figure 6 shows the diagram of the DAE proposed in this paper.

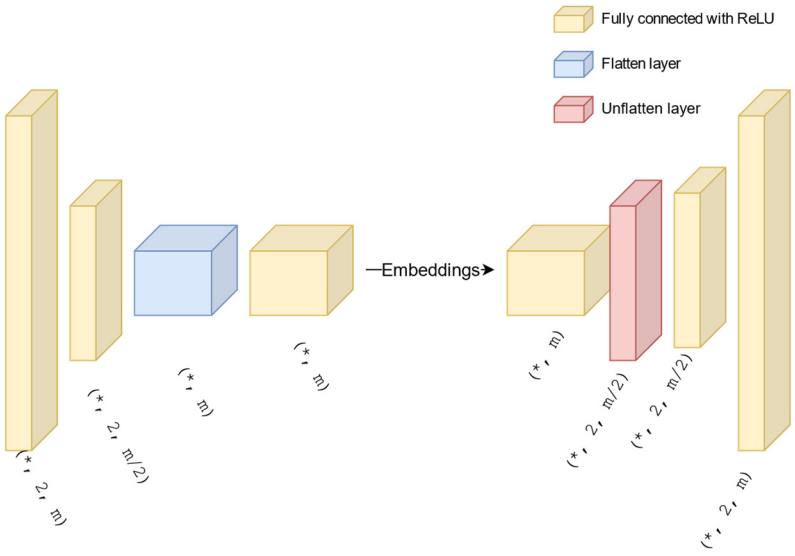

Figure 6. The architecture of the DAE proposed in this paper. The tuples in the parentheses indicate the shape of the tensors being processed by a layer, * indicates an arbitrary-sized batch dimension.

This DAE accepts as an input a tensor with shape $(*, 2, 8)$, compresses it into an embedding tensor with shape $(*, 8)$ and then restores the original shape while removing the noise from the input data. Here $*$ indicates an arbitrary size of the first dimension, which is the batch size dimension. This DAE accepts eight sequential measurements for both the SWH and the WS. This number of measurements was selected because the dataset contains three-hourly measurements and eight measurements cover 24 h. As one can see, lead time does not apply to this DAE, as it does not use the history of measurements while predicting the target value. Instead, it accepts measurements for eight sequential time points and returns the denoised measurements for the same time points. Therefore, in this paper, the DAE implements sequence-to-sequence encoder–decoder architecture to correct the prediction of a numerical model.

As one can see, the DAE accepts and returns two-dimensional data for a single object (an additional third dimension is added to make the DAE process multiple objects at a time) but uses only one-dimensional object embeddings. This means that the DAE uses the same embedding value to denoise both the SWH and the WS. Since the SWH and the WS are assumed to be closely interconnected and that they can affect each other, it is possible to argue that combining the information about both of the parameters into a single vector can benefit the DAE's ability to denoise the data.

As with GBT, two DAEs have been trained: the first for the Bares buoy and the second for the Faro buoy. No hyperparameter optimisation was performed because the architecture of DAE networks is usually determined by the number of input values they accept and, therefore, is not a subject of optimisation. Both of the DAEs have been trained using the Adam optimiser [39], with the learning rate set to 0.001. Table 6 shows the metric values computed for the DAEs using the test dataset.

Table 6. Metrics computed for the test dataset using the DAE models.

Metric	Bares Buoy		Faro Buoy	
	SWH	**WS**	**SWH**	**WS**
MSE	0.5186	3.6433	1.0145	4.0715
MAE	0.5551	1.5029	0.7885	1.5692
NRMSE	0.1927	0.2289	0.4695	0.3243

As one can see from Table 6, for the Bares buoy the DAE managed to improve the WS prediction notably; however, for the SWH the results are poor. For the SWH, the DAE did

not manage to improve the numerical model result. This may be because the results of the numerical model for the SWH are already of very good quality; therefore, the DAE is not able to tell the noise apart from the clear value.

In the case of Faro, the DAE demonstrates poor results for both the SWH and the WS. Note that for the Faro buoy, the training dataset contains almost two times less data than for the Bares buoy. Therefore, the DAE may be unable to capture the noise patterns from such a small dataset. Figure 7 shows the comparison between the actual buoy data, the numerical model prediction and the corrected numerical model prediction.

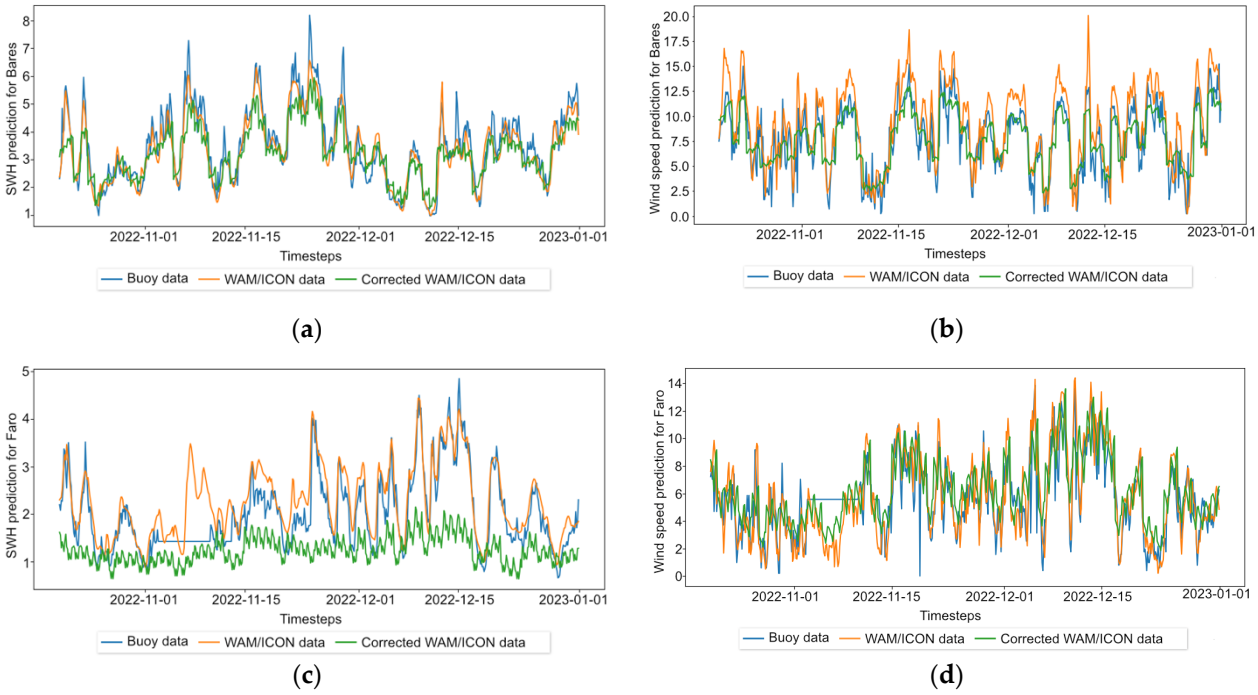

Figure 7. Comparison between the buoy data, numerical model prediction and the numerical model prediction, corrected with the DAE. (**a**) Corrected WAM/ICON data for the SWH for the Bares buoy. (**b**) Corrected WAM/ICON data for the WS for the Bares buoy. (**c**) Corrected WAM/ICON data for the SWH for the Faro buoy. (**d**) Corrected WAM/ICON data for the WS for the Faro buoy.

As one can see from the plots, in the case of the WS for the Bares buoy, the corrected prediction is almost always much closer to the buoy measurements than the non-corrected numerical model prediction (Figure 7b). However, near the peaks, the DAE smoothens the prediction. This can be explained by the hypothesis that the DAE considers peak values as more noisy and thus tries to remove more noise. In the case of the SWH for the Bares buoy, the DAE managed to capture the stochastic nature of the process, but the corrected values have a smaller magnitude than needed, even though they tend to follow the same growth-reduction pattern as the actual value. This supports the hypothesis that the DAE is not able to distinguish noise from the data and considers the data to be noisier than it actually is.

In the case of the SWH parameter for the Faro buoy, one can see from Figure 7c that the DAE was not able to properly capture the patterns in the data. The DAE is able to capture the stochastic nature of the process and sometimes follows its growth-reduction pattern, but the overall performance is poor. For the WS of the Faro buoy, the DAE seems to have captured the patterns of the original noisy data. This supports the hypothesis that for the Faro buoy, the small size of the dataset reduced the DAE's performance. In the case of the WS, the dataset has enough data to capture the process, but this amount of data is insufficient to distinguish noise from the data.

5. Discussion

It is not possible to compare the results obtained in this research with those presented in other papers dedicated to the same topic. This is because the existing papers are mostly dedicated to predicting the SWH and the WS or assimilating wave buoy data into the predictions [40,41] instead of correcting the numerical model's prediction for these two parameters. However, it is possible to compare the proposed methods to each other. Table 7 shows the metrics achieved by the proposed models compared to the baseline metrics.

Table 7. Metrics computed for the test dataset using the GBT and DAE models.

	GBT				**DAE**				**Baseline**			
	Bares		**Faro**		**Bares**		**Faro**		**Bares**		**Faro**	
	SWH	**WS**	**SWH**	**WS**	**SWH**	**WS**	**SWH**	**WS**	**SWH**	**WS**	**SWH**	**WS**
MSE	0.1968	2.6523	0.1262	2.3633	0.5186	3.6433	1.0145	4.0715	0.2132	5.2553	0.2062	2.5962
MAE	0.3413	1.2254	0.2537	1.1774	0.5551	1.5029	0.7885	1.5692	0.3528	1.8607	0.3332	1.1852
NRMSE	0.1187	0.1953	0.1656	0.2471	0.1927	0.2289	0.4695	0.3243	0.1235	0.2749	0.2117	0.2590

As one can see from Table 7, GBT managed to improve the numerical model's results for both of the parameters and for both of the buoys. The DAE improved the numerical model's results only for the WS at Bares buoy. Note that GBT uses all the parameters produced by the numerical model as inputs, while the DAE was only trained on the SWH and the WS. Moreover, the DAE was trained to predict and remove the residue, internally correcting the values. Therefore, the results achieved by the DAE when correcting the error for the WS of the Bares buoy demonstrate that DAEs can be used to solve the required task when there is enough data to capture the noise patterns and when the residue is significant enough to be distinguished from the data.

This paper uses two entirely different algorithms to solve the task: GBT and DAE. These algorithms use semantically different sets of input values, so the question of which algorithm is better is mainly affected by the available data. This paper recommends using the DAE when the dataset only contains the predictions of the wind and wave parameters of interest. GBT is expected to provide better results if more parameters are available (like in Table 1).

The novelty of this paper can be described as follows:

- This paper demonstrates that gradient boosted trees can be used to predict the residue between the numerical model's output and the actual buoy data, using the other numerically modelled spectral characteristics of the wind and wave processes.
- This paper analyses the importance of the wave parameters using the SHAP values, thus demonstrating which of the numerical model's outputs are the most important in correcting the model's errors.
- To the authors' knowledge, this is the first paper that represents the residue between the numerical model's outputs and the actual buoy data as noise. It also uses denoising autoencoders to correct the prediction of the SWH and the WS by a numerical model.
- This paper demonstrates that autoencoders can be effectively trained using the wave parameter data, thus producing embeddings that represent the underlying patterns of the wave process.

However, there are still ways to improve the results. Firstly, to improve the effectiveness of DAEs in the case of small datasets, one can use synthetic data. This synthetic data can be generated by adding random noise to the buoy data, thus producing surrogate noisy data. However, such synthetic noise must be drawn from the same distribution as the real one, but the real noise distribution may be unknown. Moreover, training a model to denoise the same dataset but with different noise may lead to overfitting, i.e., the ability of the model to denoise only the known data. Secondly, the embeddings produced by the

DAEs can be used as inputs to other models that either solve different tasks or solve the same task in a different way. For example, the embeddings can be used to predict the future parameters of the wave process instead of correcting the outputs of a numerical model. These two aspects are subject to future research.

6. Conclusions

This paper presents two models to correct the numerical model's prediction of the SWH and the WS for two distinct geographical locations.

The first one is a gradient boosted trees ensemble that is trained to correct the SWH and the WS predictions using the other wave parameters numerically modelled for the same time step. This model significantly improves the WS prediction for both the Bares and Faro buoys and the SWH prediction for these buoys. The improvement for the SWH is not significant because this parameter is already very good quality. The paper also uses SHAP values to investigate the influence each of the model inputs has on the model's output, demonstrating that for different parameters and different locations, different wave parameters have different influences and that the parameter being corrected is not always the most important one to estimate the error associated with it.

The second model assumes that the residue between the numerical model's prediction and the actual buoy data is noise and uses a denoising autoencoder to remove this noise. Although the performance of the DAE is poor compared to gradient boosted trees, the DAE still managed to significantly improve the results of the numerical model when predicting the WS for the Bares buoy. This DAE can also produce embeddings that properly represent all of the information needed to model wind and wave processes as a small vector. These embeddings can be used to solve other tasks; for example, they can be used to predict the future values of the parameters.

Author Contributions: Conceptualisation, I.Y. and C.G.S.; methodology, I.Y. and C.G.S.; software, I.Y.; writing—original draft preparation, I.Y.; writing—review and editing, C.G.S.; supervision, C.G.S.; project administration, C.G.S.; funding acquisition, C.G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This study has been performed under the project "WAVEFAI—Operational Wave Forecast using Artificial Intelligence", (http://doi.org/10.54499/CIRCNA/OCT/0300/2019). which is funded by the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia—FCT) under contract CIRCNA.OCT.0300.2019_1801P.01023. This work contributes to the Strategic Research Plan of the Centre for Marine Technology and Ocean Engineering (CENTEC), which is financed by FCT under contract UIDB/UIDP/00134/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used can be obtained by request to Deutscher Wetterdienst.

Acknowledgments: The authors are indebted to Dina Silva and Mariana Ré, who prepared the data sets analysed in this work.

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

References

1. Tolman, H.L. A third-generation model for wind waves on slowly varying, unsteady, and inhomogeneous depths and currents. *J. Phys. Oceanogr.* **1991**, *21*, 782–797. [CrossRef]
2. WAMDI Group. The WAM model—A third generation ocean wave prediction model. *J. Phys. Oceanogr.* **1988**, *18*, 1775–1810. [CrossRef]
3. Booij, N.; Ris, R.C.; Holthuijsen, L.H. A third-generation wave model for coastal regions, 1, Model description and validation. *J. Geophys. Res.* **1999**, *104*, 7649–7666. [CrossRef]
4. Cavaleri, L.; Abdalla, S.; Benetazzo, A.; Bertotti, L.; Bidlot, J.R.; Breivik, Ø.; van der Westhuysen, A.J. Wave modelling in coastal and inner seas. *Prog. Oceanogr.* **2018**, *167*, 164–233. [CrossRef]

5. Campos, R.M.; Guedes Soares, C. Comparison and Assessment of Three Wave Hindcasts in the North Atlantic Ocean. *J. Oper. Oceanogr.* **2016**, *9*, 26–44. [CrossRef]
6. Perera, L.P.; Guedes Soares, C. Weather Routing and Safe Ship Handling in the Future of Shipping. *Ocean. Eng.* **2017**, *130*, 684–695. [CrossRef]
7. Vettor, R.; Guedes Soares, C. Development of a ship weather routing system. *Ocean Eng.* **2016**, *123*, 1–14. [CrossRef]
8. Grifoll, M.; Martorell, L.; Castells, M.; Martínez de Osés, F.X. Ship weather routing using pathfinding algorithms: The case of Barcelona—Palma de Mallorca. *Transp. Res. Procedia* **2018**, *33*, 299–306. [CrossRef]
9. Grifoll, M.; Martínez de Osés, F.X.; Castells, M. Potential economic benefits of using a weather ship routing system at Short Sea Shipping. *WMU J. Marit. Aff.* **2018**, *17*, 195–211. [CrossRef]
10. Tolman, H.L. Practical Wind Wave Modeling. In Proceedings of the Conference "Water Waves: Theory and Experiment", Howard University, USA, 13–18 May 2008; Water Waves: Theory and Experiment; World Scientific: Singapore, 2010; pp. 79–92.
11. James, S.C.; Zhang, Y.; O'Donncha, F. A machine learning framework to forecast wave conditions. *Coast. Eng.* **2018**, *137*, 10. [CrossRef]
12. Feng, X.; Ma, G.; Su, S.-F.; Huang, C.; and Boswell, M.K.; Xue, P. A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan. *Ocean Eng.* **2020**, *211*, 11. [CrossRef]
13. Jing, Y.; Zhang, L.; Hao, W.; Huang, L. Numerical study of a CNN-based model for regional wave prediction. *Ocean Eng.* **2022**, *255*, 111400. [CrossRef]
14. Campos, R.M.; Krasnopolsky, V.; Alves, J.-H.G.M.; Penny, S.G. Non-linear Wave Ensemble Averaging in the Gulf of Mexico Using Neural Networks. *J. Atmos. Ocean Technol.* **2019**, *36*, 113–127. [CrossRef]
15. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; p. 775.
16. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. *Dive into Deep Learning*; Cambridge University Press: Cambridge, UK, 2024.
17. Elbisy, M.S.; Elbisy, A.M.S. Prediction of significant wave height by artificial neural networks and multiple additive regression trees. *Ocean Eng.* **2021**, *130*, 10. [CrossRef]
18. Hu, H.; van der Westhuysen, A.J.; Chu, P.; Fujisaki-Manome, A. Predicting Lake Erie wave heights using XGBoost and LSTM. *Ocean Model.* **2021**, *164*, 23. [CrossRef]
19. Ahn, S.; Tran, T.D.; Kim, J. Systematization Systematisation of short-term forecasts of regional wave heights using a machine learning technique and long-term wave hindcast. *Ocean Eng.* **2022**, *264*, 14. [CrossRef]
20. Gao, Z.; Liu, X.; Yv, F.; Wang, J.; Xing, C. Learning wave fields evolution in North West Pacific with deep neural networks. *Appl. Ocean. Res.* **2023**, *130*, 103393. [CrossRef]
21. Puscasu, R.M. Integration of Artificial Neural Networks into Operational Ocean Wave Prediction Models for Fast and Accurate Emulation of Exact Non-linear Interactions. *Procedia Comput. Sci.* **2014**, *29*, 1156–1170. [CrossRef]
22. Browne, M.; Castelle, B.; Strauss, D.; Tomlinson, R.; Blumenstein, M.; Lane, C. Near-shore swell estimation from a global wind-wave model: Spectral process, linear, and artificial neural network models. *Coast. Eng.* **2007**, *54*, 445–460. [CrossRef]
23. Londhe, S.N.; Shah, S.; Dixit, P.R.; Balakrishnan Nair, T.M.; Sirisha, P.; Jain, R. A Coupled Numerical and Artificial Neural Network Model for Improving Location Specific Wave Forecast. *Appl. Ocean Res.* **2016**, *59*, 483–491. [CrossRef]
24. Fan, S.; Xiao, N.; Dong, S. A novel model to predict significant wave height based on long short-term memory network. *Ocean Eng.* **2020**, *205*, 13. [CrossRef]
25. Pirhooshyaran, M.; Snyder, L.V. Forecasting, hindcasting and feature selection of ocean waves via recurrent and sequence-to-sequence networks. *Ocean Eng.* **2020**, *207*, 14. [CrossRef]
26. Costa, M.O.; Campos, R.M.; Guedes Soares, C. Enhancing the accuracy of metocean hindcasts with machine learning models. *Ocean Eng.* **2023**, *287*, 13. [CrossRef]
27. Zängl, G.; Reinert, D.; Rípodas, P.; Baldauf, M. The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Q. J. R. Meteorol. Soc.* **2015**, *141*, 563–579. [CrossRef]
28. Dobrynin, M.; Reinert, D.; Prill, F.; Zängl, G.; Sievers, O.; Bruns, T.; Günther, H.; Behrens, A. ICON-waves: Towards an atmosphere-waves coupled coupled operational system at DWD. In Proceedings of the DACH2022, Leipzig, Germany, 21–25 March 2022. DACH2022-167. [CrossRef]
29. Wetter und Klima-Deutscher Wetterdienst-Leistungen-legend_ICON_wave_EN_opendata.pdf. 2017. Available online: https://www.dwd.de/DE/leistungen/opendata/help/modelle/legend_ICON_wave_EN_pdf.pdf (accessed on 18 April 2024).
30. Campos, R.M.; Costa, M.O.; Almeida, F.; Guedes Soares, C. Operational wave forecast selection in the Atlantic Ocean using Random Forests. *J. Mar. Sci. Eng.* **2021**, *9*, 298. [CrossRef]
31. Kearns, M.J.; Valiant, L.G. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *J. ACM* **1994**, *41*, 67–95. [CrossRef]
32. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *arXiv* **2017**, arXiv:1706.09516.
33. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794.
34. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A highly efficient Gradient Boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.

35. Gulati, P.; Sharma, A.; Gupta, M. Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review. *Int. J. Comput. Appl.* **2016**, *141*, 19–25. [CrossRef]
36. Lundberg, S.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *arXiv* **2017**, arXiv:1705.07874.
37. Wang, H.; Liang, Q.; Hancock, J.T.; Khoshgoftaar, T.M. Feature selection strategies: A comparative analysis of SHAP-value and importance-based methods. *J. Big Data* **2024**, *1*, 16. [CrossRef]
38. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising auto-encoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; p. 8.
39. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representation, San Diego, CA, USA, 7–9 May 2015; p. 15.
40. Rusu, L.; Guedes Soares, C. Impact of assimilating altimeter data on wave predictions in the western Iberian coast. *Ocean Model.* **2015**, *96*, 126–135. [CrossRef]
41. Jiang, H.; Zhang, Y.; Qian, C.; Wang, X. Comment on papers using machine learning for significant wave height time series prediction: Complex models do not outperform auto-regression. *Ocean Model.* **2024**, *189*, 7. [CrossRef]

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