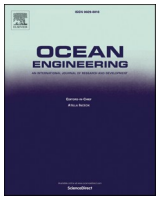
Ocean Engineering 266 (2022) 113037

Contents lists available at ScienceDirect

Ocean Engineering

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



A significant wave height forecast framework with end-to-end dynamic modeling and lag features length optimization

Hengyi Yang [a], Hao Wang [a], Yiyue Gao [a,b], Xiangyu Liu [a,c], Minyi Xu [a,*]

[a] Dalian Key Lab of Marine Micro/Nano Energy and Self-Powered Systems, Marine Engineering College, Dalian Maritime University, Dalian, 116026, China
[b] School of Science, Dalian Maritime University, Dalian, 116026, China
[c] State Key Laboratory of Physical Chemistry of Solid Surfaces, Collaborative Innovation Center of Chemistry for Energy Materials (iChEM), Tan Kah Kee Innovation Laboratory, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, 361005, China

ARTICLE INFO

Keywords:
Significant wave height
Automated machine learning
Bayesian optimization
Rolling window size
Lag features length
Time series forecast

ABSTRACT

Ocean wave energy is attracting more and more attention from researchers on observation of its clean and sustainable properties. Significant wave height (SWH) is one of the key wave parameters, making accurately forecasting the SWH important for coastal/ocean engineers. In this paper, we propose a new framework for forecasting the SWH. The data were decomposed and reconstructed, and the lag features length of the input data were adaptively optimized using the Bayesian optimization (BO) algorithm. A new paradigm for end-to-end dynamic modeling (EEDM) forecast is then proposed, where data are modeled and forecasted separately for buoys at various geographical locations, with automated machine learning (AutoML) as the back-end modeling support for the paradigm. It was also trained and tested on nine buoys from NOAA National Data Buoy Center, which are located at sites with different water depths. The results show that the forecast framework has provided reliable forecasts. We also discussed the reasons for the buoy with the worst forecast in terms of model interpretability and data quality. Finally, we compared three deep learning models (simple recurrent network, long short-term memory and gate recurrent unit) and three machine learning models (principal component regression, support vector machine and K-nearest neighbor). The comparisons indicate that the AutoML turns out to be the best.

1. Introduction

The ocean is a treasure trove of resources for sustainable development, with a variety of renewable energy sources such as wave energy, wind energy, tidal energy and ocean thermal energy. With the characteristics of high energy density and good consistency, wave energy is considered as a promising renewable energy in the ocean. Significant wave height (SWH), mean wave period (MWP), and mean wave direction (MWD) are important parameters defining wave characteristics. Accurate forecast of the SWH is of great significance to marine industry and ocean engineering. Wave (along with wind) is the dominating power, which determines the design and safety of almost all marine structures (especially floating platforms). Metaocean forecast will feed reliable input data to the dynamics. For wave energy density development, the SWH is basically the most important parameters for not only the survivability but also the efficiency. Inputting parameters such as the SWH can perform theoretical calculation and trend prediction of wave energy (Yang et al., 2022). Wave energy converter (WEC) needs to convert as much mechanical energy from the environment into electricity as possible (Wu et al., 2022). The SWH information can adjust the operating state of the WEC and improve the power generation efficiency.

The current methods for forecasting the SWH can be categorized into hard methods (empirical, numerical models), soft methods (machine learning, deep learning) and hybrid methods. Initially, numerical forecasts were based upon energy balance equations and numerical computations to build wave forecast models. The trending models in recent years include the WAVEWATCH III model (Tolman, 2009, p. 14) and the simulation waves near shore (SWAN) model (Booij et al., 1999). For example, Umesh et al. proposed SWAN-SWASH framework to forecast wave height in the Bay of Bengal region (Umesh and Behera, 2021). Panfilova et al. applied the WAVEWATCH III model to numerically simulate waves in the Persian Gulf (Panfilova et al., 2021). At the same time, local meteorological centers have also launched numerical models/systems. For example, the Met Office uses the Nucleus for European Modelling of the Ocean (NEMO) community model for ocean forecasting (Storkey et al., 2010). The European Centre for Medium-Range Weather Forecasts (ECMWF) launched the

* Corresponding author.
E-mail address: xuminyi@dlmu.edu.cn (M. Xu).

https://doi.org/10.1016/j.oceaneng.2022.113037
Received 19 June 2022; Received in revised form 7 October 2022; Accepted 27 October 2022

fifth-generation real-time seasonal forecasts system (SEAS5) for atmospheric and ocean forecasting (Johnson et al., 2019, p. 5). Japan Meteorological Agency/Meteorological Research Institute Launches Coupled Prediction System version 2 (JMA/MRI-CPS2) for Atmosphere-Land-Ocean (Takaya et al., 2018). Such methods, while achieving more accurate forecasts in some seas, require highly sophisticated computational resources and more reliable wind farm data as input.

Machine learning is a data-driven approach that focuses only on the data relationships that exist between inputs and outputs. The ability of nonlinear mapping and adaptive learning allows machine learning to demonstrate their advantages. For instances, Scott et al. proposed to forecast the SWH using a machine learning framework instead of the SWAN numerical model (James et al., 2018). Demetriou et al. developed a hierarchical machine learning classification method to short-term forecast the SWH in the southern coasts of Cyprus region (Demetriou et al., 2021). Buenoa et al. forecasted the SWH using an extreme learning machine model optimized by genetic algorithm (Cornejo-Bueno et al., 2016).

Conventional machine learning usually requires complex feature engineering, and it can only perform single-step forecast when external forecast strategies are not used (Bontempi et al., 2012). Deep learning is a subcategory of machine learning that supports a sequence-sequence input-output structure and thus supports direct output of multi-step forecast. It also has a strong advantage of feature extraction, yet it requires a large amount of data to improve the learning ability. For instances, Fan et al. used long short-term memory network for multi-step forecast of the SWH and combined it with the numerical model SWAN (Fan et al., 2020). Quach et al. employed deep neural network to forecast the SWH from synthetic aperture radar and compared it with data from the altimeter (Quach et al., 2021). Yang et al. forecasted the SWH data based on convolutional neural network and introduced position coding (Yang et al., 2021). All these SWH forecast models usually yield a single algorithm/model, i.e., by generalizing the model performance to face a large number of study subjects (buoy sites).

However, there is no single algorithm that can consistently maintain the best performance across all different sites (Sun et al., 2021). This can be attributed to two reasons. One is that the data possesses spatial heterogeneity (data features change in time and space), and the other is that a single model itself has limited hard-architectural capabilities (even after hyperparameters optimization). In face of large numbers of sites (e.g., forecasts on a global scale), it is difficult for a single algorithm research paradigm to meet the large-scale training performance.

Compared with machine learning and deep learning, automated machine learning (AutoML) is a novel modeling paradigm that automates the construction of multiple algorithmic models for training and hyperparameter optimization within a given computational resource and returns the best model (He et al., 2021). The advantages of high automation and the possession of multiple model candidates have led to its gradual application in other fields. Sun et al. reconstructed the total GRACE water storage by H2O-AutoML (Sun et al., 2021). Guo et al. analyzed urban flood warning by using tree-based pipeline optimization tool (Guo et al., 2022). Zeng et al. constructed a method to identify invasive ductal carcinoma based on Google cloud AutoML (Zeng and Zhang, 2020).

In addition, for the data-driven model described above, the quality of the input data can affect the model performance. Incorporating irrelevant features into training, or expelling critical features from training can affect the training of the model. For example, in machine learning, smoothed data after filtering and denoising could usually improve the learning ability of the model. In deep learning, feature scaling (normalization or standardization) is performed on the data to better fit the model or to optimize the algorithm's preferences. For example, data normalization can speed up the convergence of gradient descent algorithms. The SWH time-series forecast can be converted to a supervised regression problem. In multi-step forecast, direct use of timestamp features does not bring valid historical information. The sliding window method is usually used to create lag features dataset, which is the historical multi-step information, as the input information. How to reasonably select the length of historical information as input has been an often neglected but critical issue. It is usually determined empirically or using fixed multiples (Huang and Dong, 2021; Mahjoobi and Adeli Mosabbeb, 2009; Rana and Rahman, 2020; Yang et al., 2021), however, this inflexible treatment tends to lead to models that do not capture valid information.

Therefore, to address the above issues, we make the following attempts: on observation of the previous modelling limitations of using a single model, we propose a new paradigm of end-to-end dynamic modeling (EEDM) for forecast, which models and predicts data separately for buoys on various geographical locations. For the models, it is also not restricted to a particular kind of algorithm, but utilizes a pool of stand-by models. Automated machine learning serves as the back-end support for the paradigm, which adaptively selects the best model algorithm based on the performance on the validation data. For the lag length of the input data, we treat it as a black-box function and apply a Bayesian optimization (BO) algorithm for automatic search to explore the value closest to the global optimum in as few attempts as possible. Finally, in terms of the SWH feature smoothing, we invoke for the first time a regression-based data decomposition approach. The paper is significant in that it constructs a workflow for forecasting the SWH using automated machine learning, emphasizing the new concept of end-to-end modeling. What's more, this paper introduces a BO algorithm to optimize the input lag features length. Section 2 describes the data sources and details of the research methodology. Section 3 presents the related results and discussion and Section 4 concludes the whole paper.

2. Methods

2.1. Data sources and pre-processing

The reanalysis data, satellite altimeter data and buoy data are the basic sources of the SWH. In this paper, we only use the measured data from the buoy, for satellite altimeter data and reanalysis data are processed by post-computation. We have chosen 9 buoys with different water depths. The buoy data were obtained from the national data buoy center (Meindl and Hamilton, 1992), accessed on March 2, 2022 (https://www.ndbc.noaa.gov). The standard meteorological data in the buoy sites include 14 variables such as significant wave height, wave energy period, sea surface temperature, and mean wind speed, and the data are recorded at 1-h intervals. Here we use only the SWH data, without interpolating the missing values, in order to use the raw buoy data as much as possible. At the same time, each buoy is regarded as an independent object, and each buoy is only allowed to obtain its own historical SWH data. The buoy geolocation and data information are presented in section 3.1 Study area.

2.2. Data reconfiguration smoothing

Data from buoy sensors are susceptible to noise. Smoothing the data can reduce the complexity in the raw data, making it easier to identify the curvilinear features of the data. STR is an R package used to decompose the data (Dokumentov and Hyndman, 2021). And it is a seasonal-trend decomposition method using regression, combining components in an additive manner. In contrast to the conventional STL (seasonal and trend decomposition using Loess) method, STR takes into account external factors affecting the seasonal pattern of the data. It is able to combine multiple external regressions into the decomposition process. The AutoSTR function in STR also supports automatic selection of the decomposition parameter lambda using the default fivefold cross-validation. The decomposition is performed using the best combination explored by AutoSTR. The data reconfiguration smoothing details are presented in section 3.2 Forecast framework.

2.3. Bayesian optimization

For a black-box optimization problem, Bayesian optimization finds the

value that minimizes or maximizes the objective function (black-box function) in less attempts. This is building alternative functions (probabilistic models) based on the results of a priori evaluation. In simple terms, Bayesian optimization are probability distribution-based, iterative running optimization algorithms. The prior function and the acquisition function are the two core processes. Prior function models are commonly used in Gaussian processes, neural networks, and generalized linear models. Gaussian processes and neural networks are non-parametric models, in which the parameters will increase with the data augmentation. Generalized linear model is parametric models with the constant parameters. For the acquisition function (which is generally constructed by the posterior distribution of the objective function), it balances the exploration (making the next sample point as far away from the known point as possible) and exploitation (making the next sample point as close to the known point as possible) of the sampling process. The commonly used acquisition functions are probability of improvement (PI), excepted improvement (EI), and GP upper confidence bound (GP-UCB). R package ParBayesianOptimization (version 1.2.4) is used to implement the Bayesian optimization process (Snoek et al., 2012; Wilson, 2021).

2.4. Automated machine learning

Automated machine learning represents a new paradigm for optimal algorithm selection, model structure selection, and hyperparameter tuning which addresses some of the most challenging problems in machine learning applications. AutoML is able to train multiple stand-by models and select the best combination performing of algorithm and parameters (hyperparameters). The automated modeling paradigm is able to take into account more data heterogeneity than using a straightforward single model, while it is difficult for a single model to achieve uniform outstanding performance across data.

In this paper, we use the H2O-AutoML algorithm (version 3.36.1.1) to build automated machine learning processes (LeDell and Poirier, 2020). Due to the fact that in recent studies it has been shown that H2O-AutoML leads in automated machine learning, especially in supervised regression problems (Ferreira et al., 2021; Truong et al., 2019). Stand-by models includes distributed random forest (DRF), extremely randomized trees (XRT), generalized linear model (GLM), gradient boosting machine (GBM), extreme gradient boosting (XGBoost), deep learning (DL) and stacked ensemble models (SE). DRF is based on a bagging integrated learning algorithm with multiple disparate and unrelated decision trees, and is often used as a classifier with high accuracy for training and forecasting samples (Breiman, 2001). XRT is a modification of DRF, which randomly selects the best bifurcation attribute instead of traversing all feature attributes and selecting the form with the largest bifurcation value to bifurcate, so the generalization ability of XRT algorithm will be higher than DRF (Geurts et al., 2006). GLM is an extension of simple least squares regression (OLS) that is able to combine several different predictor variables to forecast the dependent variable, directly exploring the quantitative relationship between the dependent and independent variables, and also relaxing the assumptions in OLS (Faraway, 2016). GBM trains weak learners to learn the mapping of features to residuals, and each new iteration is reducing the residuals from the previous iteration, which is similar to gradient descent, so that the model will proceed in the direction of the fastest residual reduction (Friedman, 2001). XGBoost is an algorithm that supports parallel computing optimized on the basis of gradient boosting decision tree (GBDT) (Chen et al., 2015). While the general boosting algorithm is to boost the underlying weak classifiers to strong classifiers, XGBoost adds a regular term related to the number of leaf nodes to the decision tree algorithm, adjusts the parameters, controls the complexity, and prevents extreme cases such as overfitting. DL is able to automatically extract features from the training sample data, but relies on the amount of the data. In H2O-AutoML, SE includes two categories, one integrates all trained models and another integrates only the best performing models of each algorithm. Given the computational resource limit (maximum training time), H2O-AutoML automatically completes the training and hyperparameter optimization of the stand-by models. The metric for model early stopping is set to mean residual deviance and the metric for final model ranking is set to mean absolute error (MAE). This is because we do not want the models to over-consider or fit certain extreme variations in the data (Yang et al., 2021).

Our previous research attempted to create a timestamped feature matrix, which was then fed into the model for training. In this way, the features obtained by the model only have timestamps. In the multi-step forecast scenario, the model needs to obtain more features, especially the historical data of the SWH. In this paper, we try to create a dataset using a sliding window method (convert time series forecast to a supervised learning problem). At the same time the raw SWH data is not subjected to any additional processing, which has the advantage of preserving the original scale and features of the SWH data (Pirhooshyaran and Snyder, 2020). Sliding window data with missing values at any point are removed. The window data after removing the missing values are sequentially split into training dataset, tuning dataset, validation dataset and test dataset in the ratio of 6:1:1:2. Among them, Tuning dataset is used for validation data for training within H2O-AutoML. It is worth noting here that there are different numbers of missing values in different buoys, so we did not divide the data directly in proportion to the length of time, which would avoid an imbalance between different kinds of data. Also, the model does not get any information about the test dataset during the training process. The test dataset is only used during model forecast and interpretation. The default K-Fold cross-validation of H2O-AutoML is not used because it is not applicable to spatiotemporal data (Pokhrel, 2021), which leads to information leakage.

2.5. Multi-step ahead forecast strategy

The step refers to the forecast range, that is, how many units of data are forecasted. To achieve multi-step ahead forecast, two basic strategies exist: recursive forecast and direct forecast (Wang et al., 2016). Define $[y_1, y_2, \ldots, y_N]$ as the original sequence of inputs, and $[y_{N+1}, y_{N+2}, \ldots, y_{N+H}]$ as the value of the forecast ahead H steps. Specifically, the essence of the recursive strategy is to convert multi-step forecast into one-step forecast, usually by training only a one-step ahead model M1. M1 keeps using the previous forecast results as the input information in the next step, and keeps rolling forward the forecast, which will lead to the accumulation of forecast errors. In equation (1), d is the embedding dimension (lag features length) and $\widehat{f}$ is the functional dependency.

$$\widehat{y}_{N+h} = \begin{cases} \widehat{f}(y_N, y_{N-1}, \ldots, y_{N-d+1}), \ if\ h = 1 \\ \widehat{f}(\widehat{y}_{N+h-1}, \ldots, y_N, \ldots, y_{N-d+h}), \ if\ h \in \{2, \ldots, d\} \\ \widehat{f}(\widehat{y}_{N+h-1}, \widehat{y}_{N+h-2}, \ldots, \widehat{y}_{N+h-d}), \ if\ h \in \{d+1, \ldots, H\} \end{cases} \tag{1}$$

The essence of the direct strategy is to build separate models for different ahead steps, which will prevent the model from effectively capturing the correlation between the upstream and downstream and consume large computational resources. The output of the H2O-AutoML algorithm only supports one target column with one number. We cannot directly output multiple steps results as in the Seq2Seq architecture. Seq2Seq is an encoder-decoder architecture that takes one structured sequence as input and another structured sequence as output, so it can directly output multi-step prediction results (Zhang et al., 2021). In consideration, we combine two strategies, i.e., using direct strategies to optimize separately on different forecast ahead steps (different single-step models for different ahead steps). Recursive strategy is used to complete multi-step forecast.

2.6. Baseline models

Three deep learning models including recurrent neural network (RNN), long short term memory network (LSTM), gated recurrent unit (GRU) were done using R packages keras (version 2.7.0.9) and

tensorflow (version 2.7.0.9) (Falbel et al., 2022; Kalinowski et al., 2022). Three machine learning models including K-Nearest neighbor (KNN), support vector machine (SVM) and principal component regression (PCR) were achieved using the R packages Rminer (version 1.4.6) (Cortez, 2020).

2.7. Statistical metrics

Root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), spearman correlation coefficient (SCC) and R-Square were used to evaluate the model performance. Related calculations were done by the R packages yardstick (version 0.0.9) and metrics (version 0.1.4) (Hamner et al., 2018; Kuhn et al., 2022). MAPE, SCC and R-Square are dimensionless metrics. The units of MAE and RMSE are the same as the raw data units. RMSE is more sensitive to the size of the error, and the smaller the value, the better the fitting effect. MAE is more tolerant of outliers than RMSE. The value range of MAPE is [0,+∞), and 0 represents a perfect model. R-Square represents the explanatory ability of the regression model, the value range is [−1, 1], the larger the value, the better the model fitting effect. The range of SCC is [−1, 1], and the larger the value, the stronger the temporal correlation between the two groups of data. Assuming that y_i is the true value, $\hat{y}$ is the forecasted value, and m is the amount of data, the mathematical formula is as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y})^2} \tag{2}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y})| \tag{3}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}}{y_i}\right| \tag{4}$$

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum(\hat{y} - y_i)^2}{\sum(y_i - \bar{y})^2} \tag{5}$$

$$SCC = 1 - \frac{6\sum d_i^2}{m^3 - m} \tag{6}$$

where, *SSE* (sum of squared residuals error) represents the sum of the squares of the difference between the true value and the forecasted value. *SST* (sum of squares total) represents the sum of the squares of the difference between the true data and its mean. d_i is the difference between the ranks of each pair of data in the two sets of data.

Shannon entropy is used to describe the uncertainty of information (Shannon, 1948), the larger the value, the more difficult to forecast the data. We use the R package tsfeatures (version 1.0.2) to calculate the spectral entropy of the SWH (Hyndman et al., 2020). The formula is as follows:

$$H_s(x_t) = -\int_{-\pi}^{\pi} f_x(\lambda) log\, f_x(\lambda) d\lambda \tag{7}$$

where, x_t is a univariate time series. $f_x(\lambda)$ is an estimate of the spectral density of the data.

2.8. Computational resources

The hardware environment is a 24-core 2.30 GHz CPU (Intel Xeon Platinum 8260L), and the memory is 120 GB. The system environment is Ubuntu 18.04, and all experiments are written in R (version 4.1.1) and other outstanding R packages. Sapply function is used to implement parallel computation for efficiency.

3. Results and discussion

3.1. Study area

As shown in Fig. 1a, in order to evaluate the performance of the

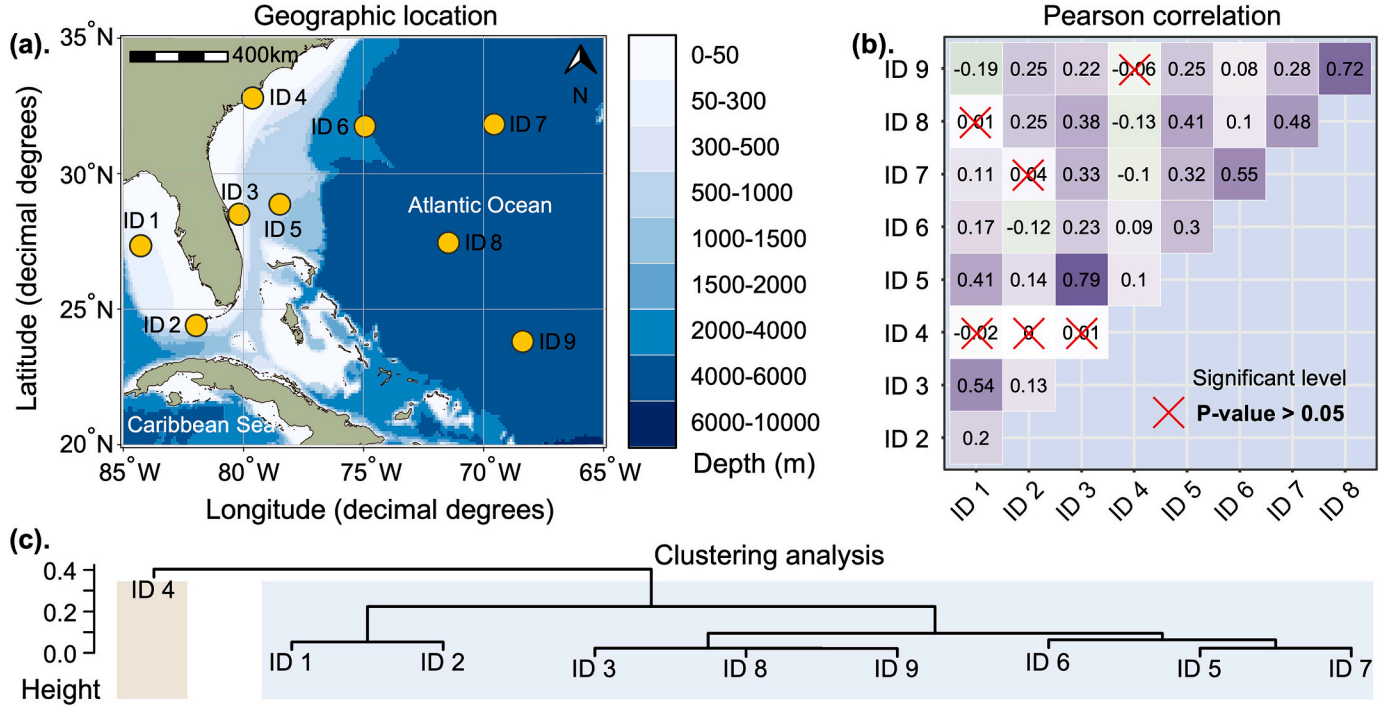


Fig. 1. (a) Visualization of buoys' geographic location, ID and water depth using R package ggOceanMaps (version 1.2.6) (Vihtakari, 2022, p. 2). (b) Pearson coefficient matrix between buoys. (c) The SWH time series clustering analysis of buoys.

Table 1
Coordinates, site code (NDBC code), water depth of buoys and the SWH statistical characteristics.

ID	Location	Site code	Min (m)	Max (m)	Mean (m)	Std (m)	Depth (m)	Data amount
1	(27°20′55″ N, 84°16′30″ W)	42099	0.13	4.13	0.90	0.55	93.9	7922
2	(24°24′26″ N, 81°58′1″ W)	42095	0.13	4.42	0.77	0.43	99.1	7784
3	(28°30′27″ N, 80°11′6″ W)	41009	0.27	3.91	1.04	0.51	42.0	8552
4	(32°48′6″ N, 79°37′8″ W)	41065	0.23	4.07	0.74	0.34	11.0	4121
5	(28°52′39″ N, 78°29′6″ W)	41010	0.49	5.31	1.50	0.68	890.0	6586
6	(31°45′35″ N, 74°56′41″ W)	41002	0.46	7.01	1.47	0.73	3784.0	4942
7	(31°49′53″ N, 69°34′23″ W)	41048	0.50	8.93	1.83	0.91	5394.0	8575
8	(27°27′48″ N, 71°27′56″ W)	41047	0.65	4.06	1.47	0.50	5347.0	5491
9	(23°49′19″ N, 68°23′2″ W)	41046	0.61	4.95	1.62	0.55	5549.0	8570

forecast framework in waves with different statistical characteristics, considering the water depth and data availability of the stations, in the spatial dimension, we selected 9 buoy stations as the research objects. In the time dimension, we focus on the SWH data in 2021 in order to assess and track the latest SWH status. The ID, coordinates, site code (NDBC code), depth and statistical characteristic information (minimum, maximum, mean, variance, data amount) of the buoys are given in Table 1.

The water depths where the buoys were located ranged from approximately 0-50 m to 4000–6000 m. To verify that the buoys possess different characteristic patterns, we calculated the Pearson coefficients among the buoys using data with common time periods. As shown in Fig. 1b, the numbers in the squares represent the Pearson coefficients. The more purple the squares are, the larger the coefficients are. Red crosses represent P values greater than 0.05 (statistical significance level), that is, they did not pass the statistical test and were not significantly different from each other.

It was able to see that there was a strong correlation between buoys ID 3 and ID 5, ID 8 and ID 9, and the correlation between the data from the other buoys was not as strong. We then performed a temporal clustering analysis using the R package TSclust (version 1.3.1) (Montero and Vilar, 2015), with the dissimilarity computation using the auto-correlation function (ACF). The clustering function hclust uses the complete linkage approach. As shown in Fig. 1c, the buoy ID 4 is different from the other 8 buoys, which may have some unique data pattern. Fig. 2 visualizes the time series of the SWH for the nine buoys. The buoys ID 4 and ID 6 start recording data on July 6, 2021 at 15:00 and June 8, 2021 at 03:00, respectively. The buoy ID 5 is vacant from July 12, 2021 at 15:00 to September 22, 2021 at 13:00. The buoy ID 8 is vacant from January 21, 2021 at 07:00 to June 6, 2021 at 01:00. Other sites have a small range of vacant data.

3.2. Forecast framework

In multi-step ahead forecast, when adding lag features to time series data (such as SWH) using the sliding window method (SWH can also be understood as spatiotemporal data, but in this paper different buoys are considered as independent objects. So, they are only considered as temporal data), all must face the problem of how to choose the length of lag features (rolling window size). This can usually be determined empirically, or through iterative trials, which will inevitably undergo a large number of training resources. The key to solving the length of lag features is how to find the global optimum with a minimum number of attempts. In this paper, we try to consider the length of lag features as a black-box function problem, that is, we only focus on the input (length of lag features) and the output (multi-step forecast performance), and its mathematical form is as follows:

$$X^* = \arg_{x \in S} \max f(x) \tag{8}$$

S is the candidate set of X. The whole expression X^* aims to maximize (or minimize) the value of $f(x)$ by selecting a suitable X from S. Where $f(x)$ is a function between the length of lag features and the model output performance, the expression of which is difficult to know. In order to solve for the point closest to the global optimum, we can first model the

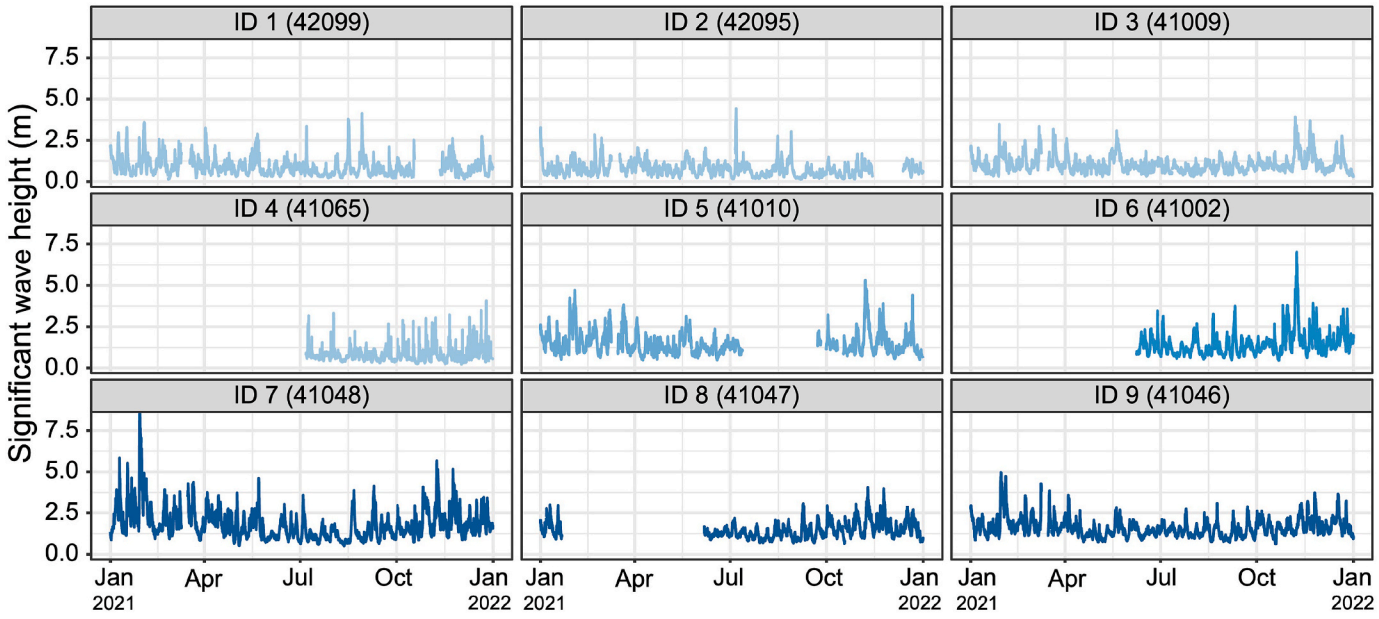


Fig. 2. The SWH time series visualization of buoys, darker color represents the greater water depth of the station. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

distribution of f(x) with a priori assumptions, and then use the subsequent feedback to continuously train the model that optimizes the conjecture to continuously approximate the f(x) function. Such methods are widely used in hyperparametric optimization in machine learning, usually grid search method, stochastic search method, genetic algorithm, particle swarm optimization method and Bayesian optimization method. Among them, genetic algorithms and particle swarm optimization belong to population optimization algorithms, which require a sufficient number of initial samples and are not efficient for optimization, and are classical black-box optimization algorithms (Kennedy and Eberhart, 1995). Grid search method and random search method are methods with general effectiveness and are not ideal in the case of limited arithmetic resources (Liashchynskyi and Liashchynskyi, 2019). Bayesian optimization constantly uses prior points to forecast posterior knowledge, so it is this feature of referring to previous evaluation results when trying the next round of hyperparameters. Bayesian optimization saves a lot of useless work and performs better (Du et al., 2022; Shields et al., 2021; Turner et al., 2021).

Therefore, we propose a new forecast framework for the SWH by solving the optimal length of lag features based on Bayesian optimization algorithm, using automated machine learning as the back-end model supporter. At the same time, the data is decomposed and reconstructed. The data processing flow is shown in Fig. 3a. The raw data are decomposed into seasonal, random and trend data. Then the random data are discarded and the remaining two data are summed to obtain the fitted data. It is worth noting that we only use it to smooth the training dataset, not to get multiple components to train/forecast individually and separately. Tuning, validation and test dataset are not smoothed and the test dataset are completely isolated. The dataset is dynamically created by Bayesian optimization algorithm. Specifically, the rolling window method is used, as shown in Fig. 4. In a sample of data, the length of lag features is the length of the training data set, and the length of the prediction data is fixed at one (H2O-AutoML only supports single-step output). Then, slide one data (step) forward, and at the new starting point, the above operations are repeated to complete the dataset creation. Simply put, the length of lag features determines how much historical data is used to predict future data.

For the Bayesian optimization process, as shown in Fig. 3b, the first step is to characterize the distribution of the objective function (green objective line) by making a prior assumption (blue proxy model), and a common conjecture in general is to assume that the objective function satisfies a Gaussian distribution (normal distribution). This is followed by an initialization process that generates a random number of initialization points (red round points), which are fed into the objective function. The objective function return value (function score) is input into the hypothesis model and utilized by the acquisition function. Then the acquisition function selects the next points (black triangle points) to evaluate in the revised model, so that the model can approach the objective function more efficiently. The acquisition functions all return unitless numbers that are used to internally quantify which points are

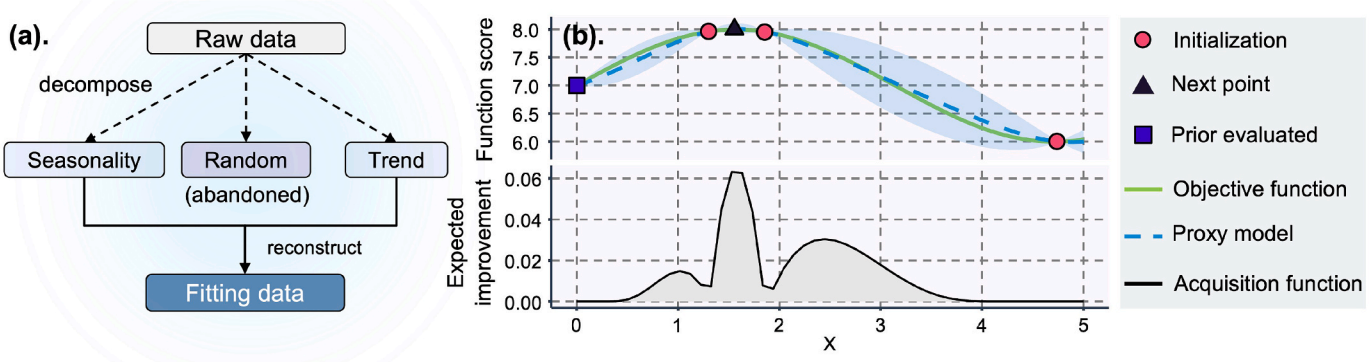


Fig. 3. (a) Smoothing process for data decomposition. (b) Visualization of Bayesian optimization process using R package mlrMBO (version 1.1.5) (Bischl et al., 2018).

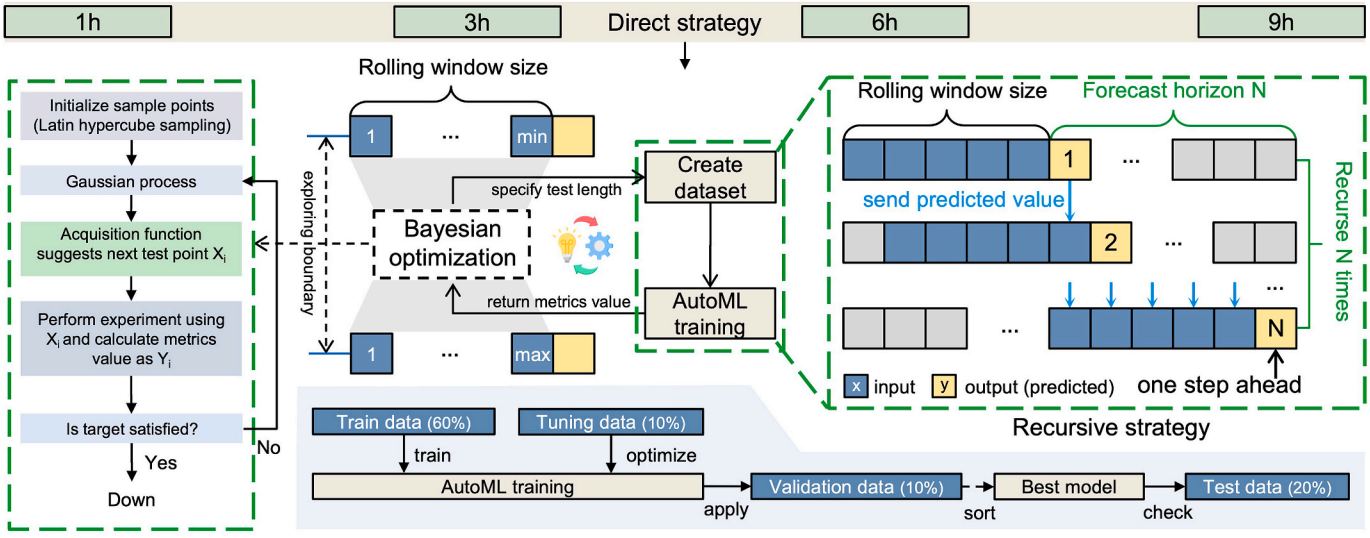


Fig. 4. Direct-Recursive forecast framework consisting of automated machine learning and Bayesian optimization of lag features length.

most exploratory. Finally, it is judged whether the new sample points satisfy the preset target, and if not, it is returned to the model as input data (blue square point) and corrected again until the target is satisfied, and then the best combination of parameters currently selected is output. In the framework, the prior function is set to Gaussian process, the acquisition function is set to expected improvement. Different forecast ahead steps are set to different boundary ranges to search in integer form. The summation values of R2,CCC and MAPE (taking opposite numbers) on the validation set are calculated as return values. For the algorithm initialization, we utilize Latin hypercube sampling, which is randomly selected within the bounded range. The metric for terminating the search is Utility, which stops optimization when the value converges to 0.

As shown in Fig. 4, in the framework, we combine two forecast strategies and optimize them separately for different ahead steps using direct strategies. Within specified ahead steps, multi-step forecast is achieved using a recursive strategy, which means that the forecast is iterated forward continuously using a single-step model. The number of forecast steps is also the number of iterations. Specifically, the forecasted values in the previous step are used as input and are sent to the next forecast step, i.e., forecasting with updates is not used because in practical applications it is impossible to know the future information in the present. 60% of the raw data is used for training, tuning data is used as validation data inside H2O-AutoML to optimize the stand-by models.

All candidate models optimized by automated machine learning are applied on the external validation dataset and the best model is selected based on MAE metrics. Finally, the performance is checked on the fully isolated test dataset.

3.3. Automated machine learning training stability

Rational setting of the maximum search time for automated machine learning can reduce unnecessary computational resources. We randomly choose 3 different lag features lengths (30 steps, 45 steps, 60 steps) and set 4 different search time (10 min, 20 min, 30 min, 40 min). Search time refers to the training time of automated machine learning. The MAPE of 24h ahead forecast is used as an evaluation metric, this is because to accumulate the most errors to check a large performance float. And the dimensionless metric is used to compare the performance between different floats. As shown in Fig. 5a, we found that the longer the search time is not better, the longer the search time may not bring much improvement. For example, buoys ID 3, 5, 6, 7 in the long search time, the performance does not get significant improvement. The rest of the sites in 10 min search time, also reached or close to the best MAPE value. So, we choose 10 min as the maximum search time for automated machine learning.

To determine whether it is necessary to try multiple training, trials were subsequently performed at the same 3 lag features lengths, with the 24h ahead steps MAPE as the evaluation metric. The automated machine learning maximum search time is set to 10 min. The results are shown in Fig. 5b. Within 5 training times, the best performance is usually achieved the first time, and even if not, the best performance is achieved within a maximum of 3 attempts. Therefore, for the subsequent formal training, we first train 2 times, check whether the performance of the two times has fluctuations, and stop trying if there are no fluctuations. Otherwise, go ahead and try a third time, and take the model with the best performance among the three repetitions as a representative. In the formal training, we choose the extreme multi-step forecast result as the floating metric on the validation dataset, and do not use any information from the test dataset.

3.4. Best model results and optimization process

Long-term forecast of the SWH (belonging to chaotic signals) is impossible (Huang and Dong, 2021). Therefore, for multi-step ahead forecast, we choose lead times of 1h, 3h, 6h, 9h. The lag features search intervals in different forecast ranges are shown in Table 2. The search boundary is the range of lag features length, and the search length is the number of different values contained in the search boundary. For different forecast horizons, we keep the search length consistent.

Since we use a hybrid direct-recursive forecast strategy, the best models differ on different forecast lead times at the same buoy site. As shown in Fig. 6a, the best models for buoys ID 2, 5, 6, and 9 remain consistent at all four ahead steps, which are SE, GLM, GLM, and GLM, respectively. The best models on buoy ID 4 vary the most, and the best models at 1h, 3h, 6h, and 9h are XGBoost, GBM, DRF, and GBM, respectively. Buoys ID 1, ID 7 have their best models change between 3h and 6h, and the best models on buoy ID 3, ID 8 only change on 6h. For the best lag features length, the buoys ID 1, 2, 4, and 9 increase the length of the lag features as the forecast ahead steps increase.

The rest of the sites are kept in a consistent or random state. Meanwhile, no one algorithm consistently performs the best. But in general, ensemble learning models and GLMs beat most algorithms. It is worth noting that machine learning models can only capture the relationship between data inputs and outputs. If there are dissimilar climate (data patterns) between the validation dataset, test dataset, and training dataset, this may yield poor decisions on the best model selection, i.e., the best model on the validation dataset may not be optimal on the test dataset. However, in any case, we cannot use the test dataset directly or indirectly to train, optimization, or even select the best model, although the results would be better.

Table 2
The boundary and length of the search lag features.

Forecast horizon	Search boundary	Search length
1h	[6,72]	67
3h	[6,72]	67
6h	[18,84]	67
9h	[18,84]	67

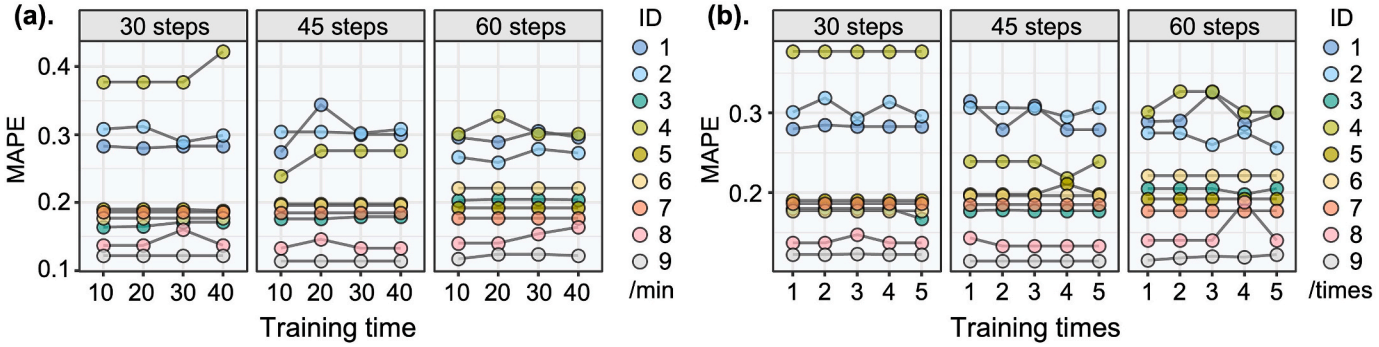


Fig. 5. Automated machine learning longest training time test (a) and training number test (b), visualized using the R package grafify (version 2.2.0) (Shenoy, 2021).

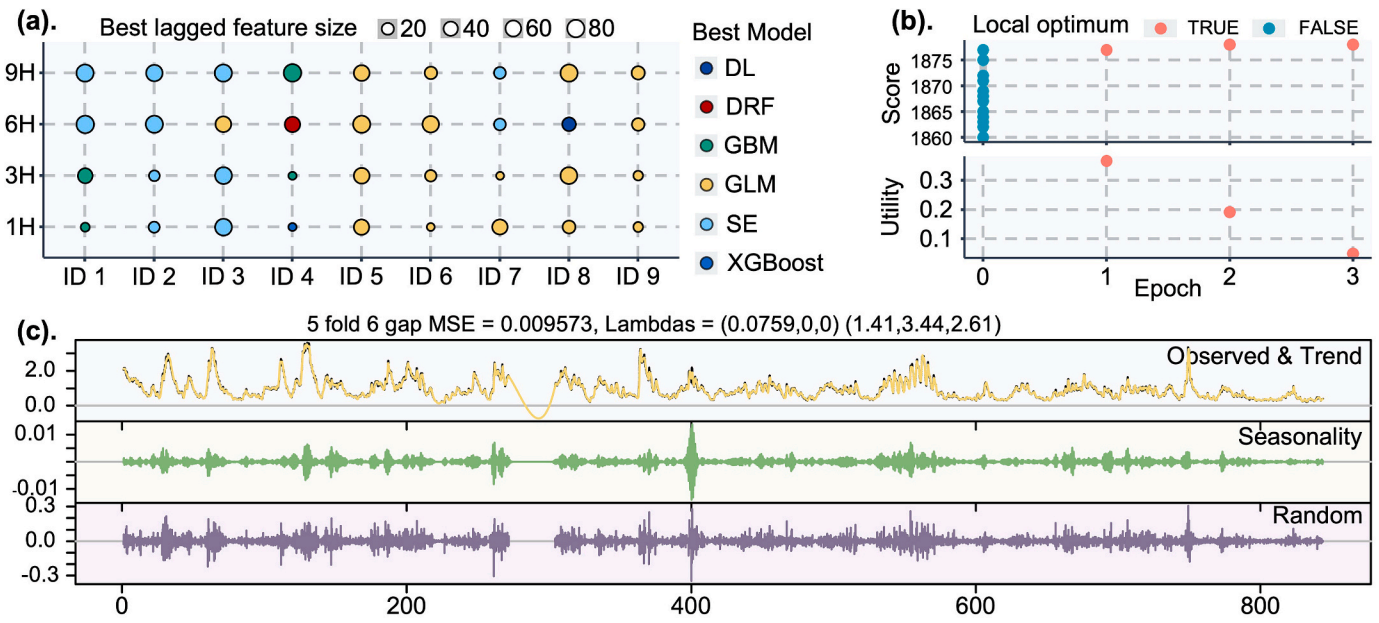


Fig. 6. (a) Results of the forecast framework on 9 buoys, visualized using the R package ggplot2 (version 3.3.5) (Gómez-Rubio, 2017). Visualization of the Bayesian optimization process (b) and decomposition-reconstruction smoothed data (c) on buoy ID 1.

Subsequently, we visualized the Bayesian optimization process on buoy ID 1. As shown in Fig. 6b, the blue points represent the initialization points that start the optimization process. The red points represent the points that the algorithm explores outward based on the previous information. The utilities represent the uncertainty of the exploration interval, when convergence to 0 means that a better point than the current parameter set has been found, i.e., the global optimum has been found. Similarly, we visualized the process of decomposing and reconstructing the training data on buoy ID 1, as shown in Fig. 6c. The black line is the original observation, the yellow line is the trend component, the green line is the seasonal component, and the purple line is the random component. The trend component and seasonal component are reconstructed into smoothed data after random variables are excluded.

3.5. Forecast performance

Fig. 7a–d show the SCC, RMSE, MAPE, and MAE of four forecast lead

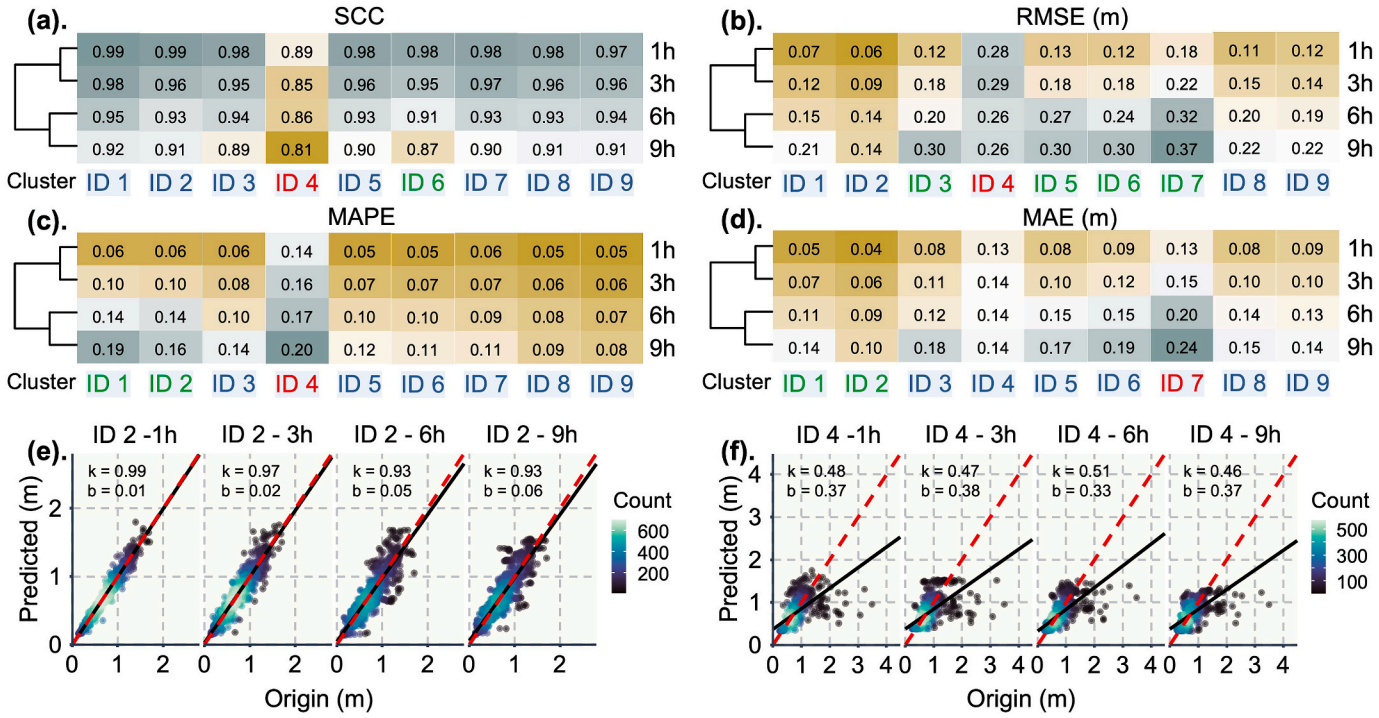


Fig. 7. Visualization of the four metrics SCC (a), RMSE (b), MAPE (c), MAE (d) on 9 buoys and different ahead steps, using the R package pheatmap (version 1.0.12) (Kolde, 2019). Scatter density maps of forecasted and true values on buoys ID 2 (e) and ID 4 (f), using the R package ggpointdensity (version 0.1.0) (Kremer and Anders, 2019).

times. The model performs best at 1h ahead step with maximum SCC of 0.99, minimum RMSE of 0.06m, minimum MAPE of 0.05, and minimum MAE of 0.04m. In the longest 9h ahead steps, the SCC is maximum 0.92, RMSE is minimum 0.14 m, MAPE is minimum 0.08, and MAE is minimum 0.10 m. Combining the results, the buoy ID 2 performed the best compared to other buoys. At the same time, we can clearly see that with the increase of the forecast ahead step, the model performance decreases to varying degrees, which is due to the accumulation of errors caused by the limitations of machine learning single-step forecast. The text color of the buoy ID represents the clustering result (the number of Clades is 3). The same color represents the same class. The clustering results for the four metrics show that the buoys clustered between ID 1 and 2, between ID 3, 5, 6 and 7, and between ID 8 and 9. This may be somehow related to the water depth where the stations are located (as in Fig. 1a). Dimensionless metrics is suitable for comparing performance between buoys. In particular, we were able to clearly observe that buoy ID 4 exhibits a MAPE and SCC performance that is clearly "unusually" poor compared to the other buoys. We speculate that this may be related to its pattern of temporal variation. As we found significant heterogeneity between ID 4 and the rest of the buoys in the buoy temporal clustering analysis (Fig. 1c). Overall, the framework forecasts are feasible in the short-term forecast range, with buoys ID 2 and 4 representing the best and worst forecast performance, respectively.

Subsequently, we looked at the situation between the forecasted and true values on these two representative buoys, as shown in Fig. 7e–f. The red dashed line is the baseline with a slope of one. The black line is the fitted straight line, k represents the slope, and b represents the intercept. In buoy ID 2, it can be seen that on the four forecasted ahead steps, a slender concentration trend is maintained between the forecasted and true values, and the slopes are all above 0.90. In contrast, in buoy ID 4, the forecasted and true values are more scattered between each other, and the fitted straight line deviates greatly from the baseline.

We then proceeded to understand the machine learning model interpretability on the test dataset of two buoys. Partial dependance describes the marginal effect of the feature variables on the model response. That is, keep other features unchanged, and then change the value of the target feature to observe the fitting of the model. As shown in Fig. 8a, among the four ahead steps on buoy ID 2, the last feature in the best lag length (referred to as the best feature) shows a high linear positive correlation with the average output of the model. The green shading represents the confidence interval. As shown in Fig. 8b, among the four forecast ahead steps on buoy ID4, the best feature does not show a simple linear relationship with the average output of the model. As the best feature increases, the average output of the model get less improvement. This may also explain why the forecast framework works better on buoy ID 2 than on ID 4. Shapley additive explanations (SHAP) summary plots combine the influence of feature importance and features. The vertical coordinates indicate the feature names, in decreasing order of importance to the model from top to bottom. Each point represents the SHAP value for each sample. The closer the color is to red, the larger its value, and the closer it is to blue the smaller its value. The more dispersed the dots in the graph, the greater the influence of the variable on the model. The best model for buoy ID 2 on all four ahead steps is the ensemble learning model, so there is no SHAP plot. As shown in Fig. 8c, the results on buoy ID 4 show that the best feature always has the greatest impact on the model. The larger the value of the best feature, the greater the output of the model, which is consistent with the results of partial dependance.

Subsequently, we visualized all the training results on the buoys ID 2 (Fig. 9a) and ID 4 (Fig. 9b). Buoy ID 4 is more sensitive than ID 2. For the effect of hysteresis feature length optimization, as shown in Fig. 9c, the overall improvement of the buoy ID 4 is higher than that of ID 2. At the same time, the larger the forecast ahead steps, the greater the improvement in performance. May because the more future information to be forecasted requires more complex historical information.

In the above results, we can see that the buoy ID 2 and 4 show a very large difference, both in performance and other connections. We guessed that this might be related to the pattern of data present on the buoy ID 4 itself, so we calculated Shannon entropy for the time series data. As shown in Fig. 9d, the results showed that the entropy of buoy ID 4 is as high as 0.80. A larger entropy means that the data are more difficult to forecast, i.e., the degree of uncertainty is higher. It qualitatively explains why the poor forecast results appear on buoy ID 4. For any forecast algorithm, the upper limit effect is dependent on the data itself or the extrinsic feature engineering (Harrington, 2012; Kläs and Vollmer, 2018). On the one hand, not all data can be forecasted (such as

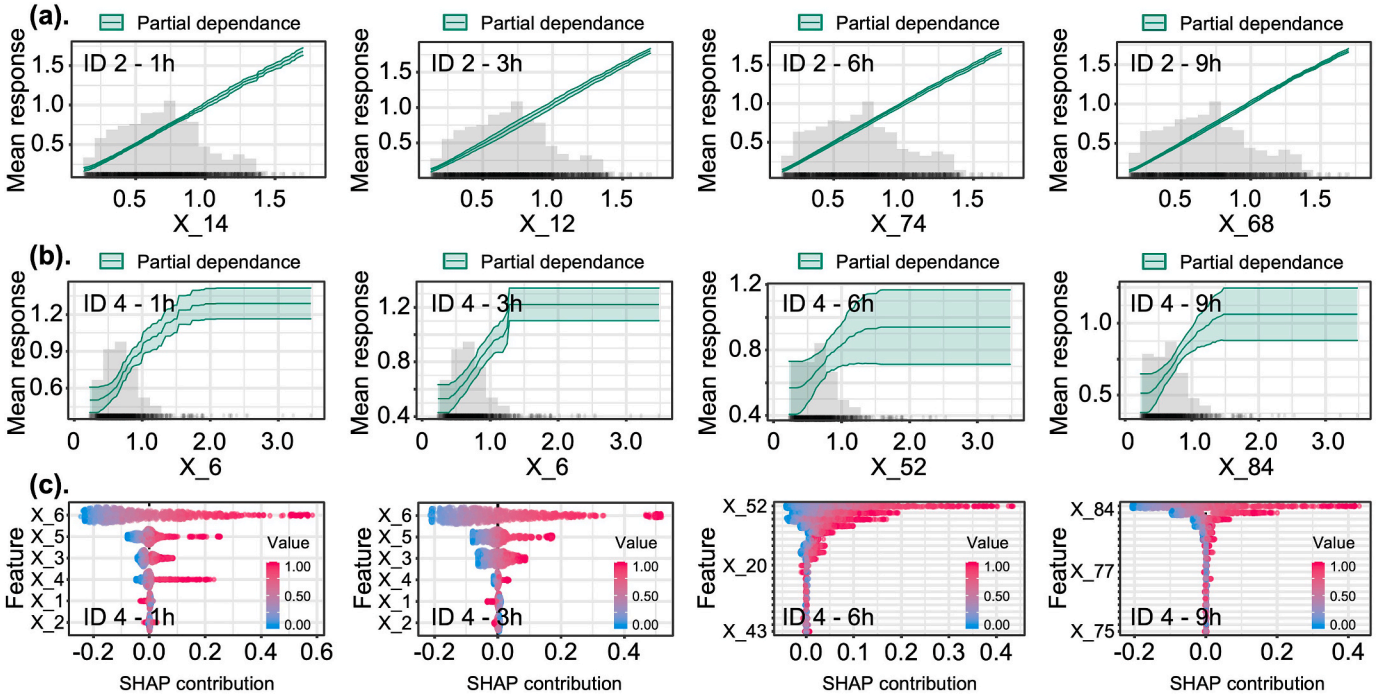


Fig. 8. Partial dependency plots of four forecast lead steps on buoys ID 2 (a) and ID 4 (b), and summary results of Shapley additive explanations on buoy ID 4 (c).

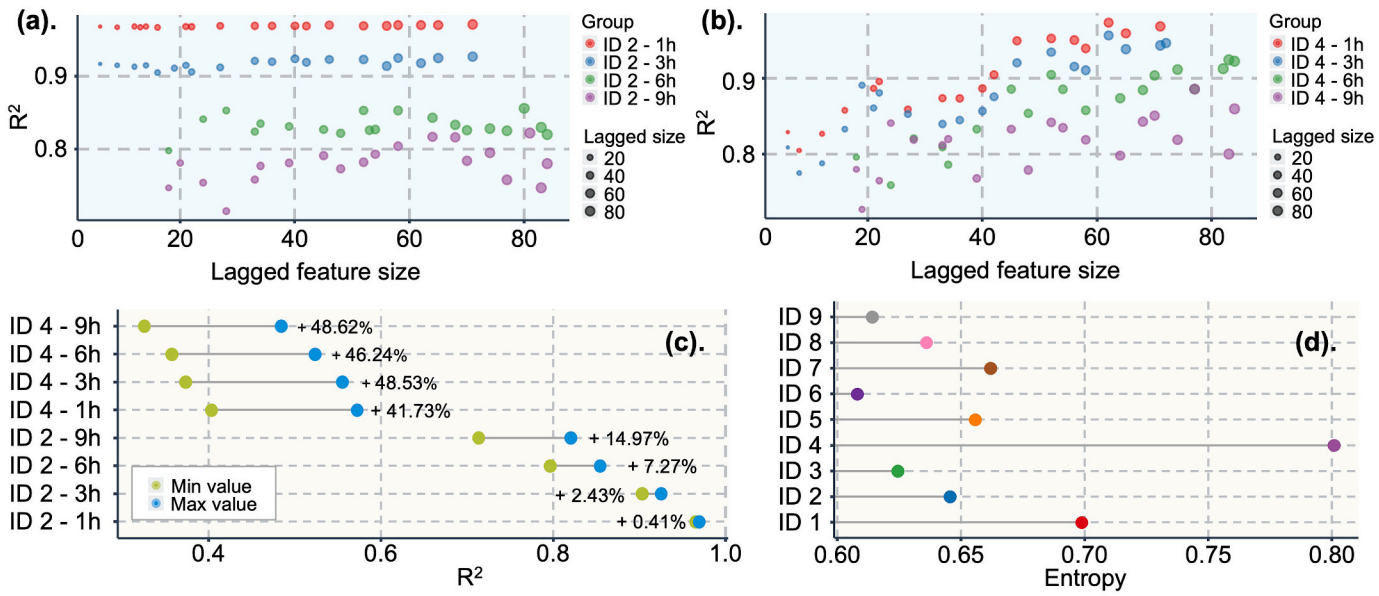


Fig. 9. All training results on buoys ID 2 (a) and ID 4 (b) and optimization boosting degree (c). (d) Shannon entropy of the SWH time series on 9 buoys.

white noise or random walk data). On the other hand, time series can be post-processed (such as removing outliers). This additional data processing is not available in the proposed framework. Here, to verify our conjecture and reveal the limitations of the framework, we post-process the data of buoy ID 4. Statistically so-called outliers (extreme values) are removed. Data processing utilizes the tsclean function of the R package forecast (Hyndman et al., 2022). It is based on STL (seasonal and trend decomposition using loess) and super smoother. And uses student-distribution based quantiles to detect outlier points. Replace outliers using linear interpolation of adjacent values. The visualization of the processed and raw data is shown in Fig. 10.

It can be seen that more statistical outliers are removed in the second half (after the middle-dotted line). The data after removing outliers is re-entered into the framework for testing, and the results are shown in Table 3:

It can be seen that after removing outliers, both R-Square (assessing the model fitting) and SCC (assessing the linear correlation of two variables) yield different degrees of improvement. As can be seen from Fig. 10, the outliers increase significantly in the second half. The framework may be overfitting on the raw buoy ID 4 data. Due to the appearance of outliers, the test and the training set have significantly different data patterns. This phenomenon is called data shift, which leads to poor generalization ability of the model. It is worth noting that the datasets size we used ranged from six months to less than one year. This is a relatively small training data and the model may not be able to capture more features. Whereas in machine learning, usually increasing the training data improves the training process of the model (Gautam and Yadav, 2014), this also needs further research. However, it is very dangerous to directly remove outliers without the user's permission. Assuming focus on the extreme physics behind outliers, users should first clarify the purpose of using the model, and then decide how to deal with outliers. Behind every outlier point is a noteworthy physical phenomenon. When regression problems (forecasting future values, continuous) do not work, we should consider converting them to classification problems (forecasting future probabilities, categorical). This paper only discusses the situation of the data itself, and the externally complex feature engineering still deserves additional discussions.

Finally, we conducted ablation experiments on the AutoML module in the forecast framework. We used six other models, three deep learning (SRN, LSTM, GRU), and three machine learning (KNN,SVM,PCR), to

Table 3
Percentage change in R-square on buoy ID 4 after removing outliers.

Metric	Data processing	ID4 - 1 h	ID4 - 3 h	ID4 - 6 h	ID4 - 9 h
R2	No	0.429	0.409	0.505	0.46
	Yes	0.796 (+85.55%)	0.718 (+75.55%)	0.706 (+39.80%)	0.666 (+44.78%)
SCC	No	0.887	0.853	0.856	0.81
	Yes	0.924 (+4.17%)	0.877 (+2.81%)	0.856	0.851 (+5.06%)

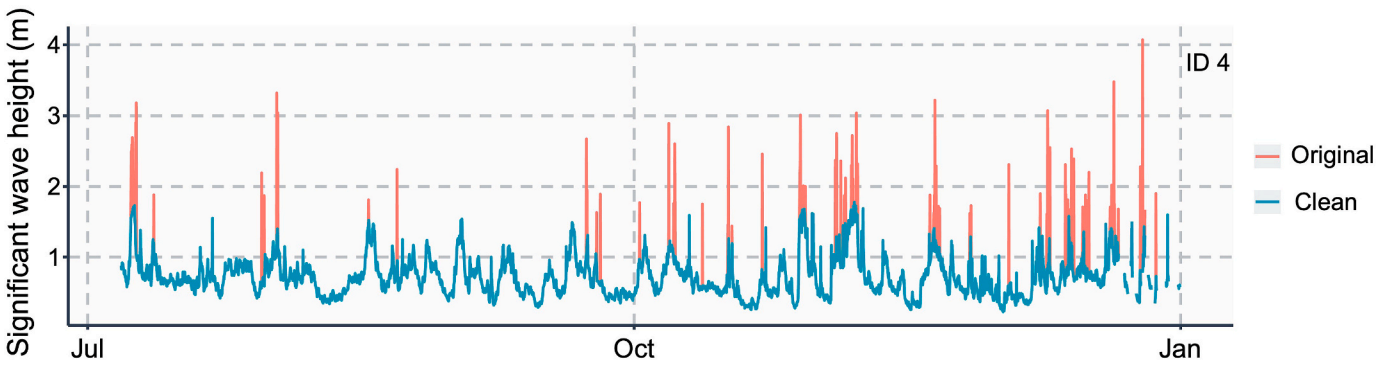


Fig. 10. Comparison of buoy ID 4 raw data (Original) and after removing outliers (Clean).

Table 4
R-Square percentage change for 6 compared models in buoy ID 2.

Frame	ID2 - 1h	ID2 - 3h	ID2 - 6h	ID2 - 9h
BO-stR-SRN	0.904 (−6.61%)	0.83 (−9.09%)	0.772 (−6.76%)	0.663 (−18.75%)
BO-stR-LSTM	0.894 (−0.72%)	0.866 (−1.42%)	0.787 (−9.06%)	0.702 (−16.54%)
BO-stR-GRU	0.921 (−4.86%)	0.867 (−5.04%)	0.772 (−6.76%)	0.709 (−13.11%)
BO-stR-PCR	0.942 (−2.69%)	0.804 (−11.94%)	0.568 (−31.40%)	0.527 (−35.42%)
BO-stR-SVM	0.965 (−0.31%)	0.9 (−1.42%)	0.778 (−6.04%)	0.672 (−17.65%)
BO-stR-KNN	0.961 (−0.72%)	0.9 (−1.42%)	0.753 (−9.06%)	0.681 (−16.54%)
BO-stR-AutoML	0.968	0.913	0.828	0.816

Table 5
R-Square percentage change for 6 compared models in buoy ID 4.

Frame	ID4 - 1h	ID4 - 3h	ID4 - 6h	ID4 - 9h
BO-stR-SRN	0.364 (−15.15%)	0.379 (−7.33%)	0.438 (−13.27%)	0.388 (−15.65%)
BO-stR-LSTM	0.399 (−6.99%)	0.375 (−8.31%)	0.396 (−21.58%)	0.447 (−2.83%)
BO-stR-GRU	0.415 (−3.26%)	0.369 (−9.78%)	0.375 (−25.74%)	0.355 (−22.83%)
BO-stR-PCR	0.063 (−85.31%)	0.05 (−87.78%)	0.001 (−99.80%)	0.014 (−96.96%)
BO-stR-SVM	0.323 (−24.71%)	0.265 (−35.21%)	0.243 (−51.88%)	0.17 (−63.04%)
BO-stR-KNN	0.348 (−18.88%)	0.294 (−28.12%)	0.393 (−22.18%)	0.346 (−24.78%)
BO-stR-AutoML	0.429	0.409	0.505	0.46

replace AutoML in the framework in buoy ID 2 (as in Table 4) and ID 4 (as in Table 5).

It is worth noting that the optimal lag features length is data and model type dependent and is not fixed. That is, for the same data, different mapping relationships between lag feature length and performance may also arise among different models. So, we form a separate forecast framework for the six models and re-explore the optimal lag features lengths, keeping the other parameters constant. The results showed that in both two buoys, the other five models have different degrees of performance degradation compared to AutoML, and magnitudes increase to different degrees as the forecast lead steps increases. The PCR in the baseline models completely failed on buoy ID 4, which may be related to model capability and data quality.

4. Conclusions

In this paper, we propose, for the first time, a SWH forecast framework using automated machine learning. The proposed framework focuses on a unique model training paradigm, a strategy that discards the previous idea of utilizing only the single algorithm and applies a pool of candidate models to adaptively select the most appropriate model with hyperparameters optimizing. The framework decomposes and reconstructs the data to reduce the influence of raw noise. Bayesian optimization algorithm is also used to select the appropriate lag features length to avoid the negative effects from using empirical data or fixed scale input data. Nine buoys with different data characteristics are used to validate the proposed framework. Compared with the case of using a single algorithm, automated machine learning can better adapt to the spatial heterogeneity of the data. Among the various buoys, automated machine learning is able to select the best model with different algorithm types. Bayesian optimization effectively improve the forecast performance of the model by optimize the lag features length. For Bayesian optimization, its own hyperparameters can also affect the optimization process, which is also worthy of further research and discussion. It is worth noting that the data structure of time series like the SWH is dynamically changing. Even if the automated machine learning outputs the best model, it may only perform well in a certain geographical location or a certain time period. We should re-evaluate the best model in the past reasonably. The poor performance of the framework on buoy ID 4 also reveals its limitations in dealing with data outliers. For the outliers, users should try to either eliminate them after clarifying their business goals or convert the problem type (from regression to classification). In future work, the SWH variables can be forecasted using a feature matrix composed of other physical quantity data. Also, the effects of spatial location should be considered and utilize adjacent buoy sites to provide additional data. The focus of future work is biased toward feature engineering of the SWH data.

Overall, this paper presents a new paradigm for the SWH forecast. It introduces the Bayesian optimization algorithm and automated machine learning, which can be considered for more applications in coastal/ocean engineering. It is also applicable for forecasting other ocean parameters in an integrated analysis.

CRediT authorship contribution statement

Hengyi Yang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing. **Hao Wang:** Writing – review & editing. **Yiyue Gao:** Investigation, Writing – review & editing. **Xiangyu Liu:** Writing – review & editing. **Minyi Xu:** Funding acquisition, Supervision, Resources, 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.

Data availability

Data will be made available on request.

Acknowledgment

The work was supported by the National Key R & D Project from Minister of Science and Technology (2021YFA1201604), the National Natural Science Foundation of China (Grant Nos. 51879022, 52101382), Project of Dalian Outstanding Young Scientific and Technological Personnel (2021RJ11). The authors acknowledge the NOAA National Data Buoy Center (NDBC) for buoy data resource, and uniconlabs for providing open-source graphic resources. Also, offer their appreciation to Zegui Deng at Tianjin University for his advice on missing value handling. The authors also thank the anonymous reviewers for their suggestions, which greatly improved the quality of the paper.

References

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., Lang, M., 2018. mlrMBO: a modular framework for model-based optimization of expensive black-box functions arXiv:1703.03373 [stat]. http://arxiv.org/abs/1703.03373.

Bontempi, G., Ben Taieb, S., Borgne, Y.-A.L., 2012. Machine learning strategies for time series forecasting. In: European Business Intelligence Summer School. Springer, pp. 62–77.

Booij, N., Ris, R.C., Holthuijsen, L.H., 1999. A third-generation wave model for coastal regions: 1. Model description and validation. J. Geophys. Res.: Oceans 104, 7649–7666.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., 2015. Xgboost: extreme gradient boosting. R package version 0 4–2 1, 1–4.

Cornejo-Bueno, L., Nieto-Borge, J.C., García-Díaz, P., Rodríguez, G., Salcedo-Sanz, S., 2016. Significant wave height and energy flux prediction for marine energy applications: a grouping genetic algorithm – extreme Learning Machine approach. Renew. Energy 97, 380–389. https://doi.org/10.1016/j.renene.2016.05.094.

Cortez, P., 2020. Rminer: Data Mining Classification and Regression Methods [WWW Document]. URL. https://CRAN.R-project.org/package=rminer. (Accessed 25 January 2022).

Demetriou, D., Michailides, C., Papanastasiou, G., Onoufriou, T., 2021. Nowcasting significant wave height by hierarchical machine learning classification. Ocean Eng. 242, 110130 https://doi.org/10.1016/j.oceaneng.2021.110130.

Dokumentov, A., Hyndman, R.J., 2021. STR: seasonal-trend decomposition using regression. INFORMS J. Data Sci. https://doi.org/10.1287/ijds.2021.0004.

Du, L., Gao, R., Suganthan, P.N., Wang, D.Z., 2022. Bayesian optimization based dynamic ensemble for time series forecasting. Inf. Sci. 591, 155–175.

Falbel, D., Allaire, J.J., RStudio, Tang, Y., Eddelbuettel, D., Golding, N., Kalinowski, T., 2022. Google Inc. tensorflow: R Interface to "TensorFlow" [WWW Document]. URL. https://CRAN.R-project.org/package=tensorflow. (Accessed 25 April 2022).

Fan, S., Xiao, N., Dong, S., 2020. A novel model to predict significant wave height based on long short-term memory network. Ocean Eng. 205, 107298 https://doi.org/10.1016/j.oceaneng.2020.107298.

Faraway, J.J., 2016. Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. Chapman and Hall/CRC.

Ferreira, L., Pilastri, A., Martins, C.M., Pires, P.M., Cortez, P., 2021. A comparison of AutoML tools for machine learning, deep learning and XGBoost. In: 2021 International Joint Conference on Neural Networks (IJCNN). Presented at the 2021 International Joint Conference on Neural Networks. IJCNN), pp. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534091.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232.

Gautam, G., Yadav, D., 2014. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. 2014 Seventh International Conference on Contemporary Computing (IC3). Presented at the 2014 Seventh International Conference on Contemporary Computing IC3, 437–442. https://doi.org/10.1109/IC3.2014.6897213.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63, 3–42. https://doi.org/10.1007/s10994-006-6226-1.

Gómez-Rubio, V., 2017. ggplot2-elegant graphics for data analysis. J. Stat. Software 77, 1–3.

Guo, Y., Quan, L., Song, L., Liang, H., 2022. Construction of rapid early warning and comprehensive analysis models for urban waterlogging based on AutoML and comparison of the other three machine learning algorithms. J. Hydrol. 605, 127367 https://doi.org/10.1016/j.jhydrol.2021.127367.

Hamner, B., Frasco, M., LeDell, E., 2018. Metrics: Evaluation Metrics for Machine Learning [WWW Document]. URL. https://CRAN.R-project.org/package=Metrics. (Accessed 25 April 2022).

Harrington, P., 2012. Machine Learning in Action. Simon and Schuster.

He, X., Zhao, K., Chu, X., 2021. AutoML: a survey of the state-of-the-art. Knowl. Base Syst. 212, 106622 https://doi.org/10.1016/j.knosys.2020.106622.

Huang, W., Dong, S., 2021. Improved short-term prediction of significant wave height by decomposing deterministic and stochastic components. Renew. Energy 177, 743–758. https://doi.org/10.1016/j.renene.2021.06.008.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F., , R Core Team, Ihaka, R., Reid, D., Shaub, D., Tang, Y., Zhou, Z., 2022. Forecast: Forecasting Functions for Time Series and Linear Models.

Hyndman, R., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y., O'Hara-Wild, M., Taieb, S.B., Hanqing, C., Lake, D.K., Laptev, N., Moorman, J.R., 2020. Tsfeatures: time series feature extraction [WWW Document]. URL. https://CRAN.R-project.org/package=tsfeatures. (Accessed 31 March 2022). accessed.

James, S.C., Zhang, Y., O'Donncha, F., 2018. A machine learning framework to forecast wave conditions. Coast. Eng. 137, 1–10. https://doi.org/10.1016/j.coastaleng.2018.03.004.

Johnson, S.J., Stockdale, T.N., Ferranti, L., Balmaseda, M.A., Molteni, F., Magnusson, L., Tietsche, S., Decremer, D., Weisheimer, A., Balsamo, G., Keeley, S.P.E., Mogensen, K., Zuo, H., Monge-Sanz, B.M., 2019. SEAS5: the new ECMWF seasonal forecast system. Geosci. Model Dev. (GMD) 12, 1087–1117. https://doi.org/10.5194/gmd-12-1087-2019.

Kalinowski, T., Falbel, D., Allaire, J.J., Chollet, F., RStudio, Google, Tang, Y., Bijl, W.V. D., Studer, M., Keydana, S., 2022. Keras: R Interface to "keras" [WWW Document]. URL. https://CRAN.R-project.org/package=keras. (Accessed 25 April 2022).

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, pp. 1942–1948. IEEE.

Kläs, M., Vollmer, A.M., 2018. Uncertainty in machine learning applications: a Practice-driven classification of uncertainty. In: Gallina, B., Skavhaug, A., Schoitsch, E., Bitsch, F. (Eds.), Computer Safety, Reliability, and Security. Springer International Publishing, Cham, pp. 431–438. https://doi.org/10.1007/978-3-319-99229-7_36.

Kolde, R., 2019. pheatmap: Pretty Heatmaps [WWW Document]. URL. https://CRAN.R-project.org/package=pheatmap. (Accessed 10 April 2022).

Kremer, L.P.M., Anders, S., 2019. Ggpointdensity: a cross between a 2D density plot and a scatter plot [WWW Document]. URL. https://CRAN.R-project.org/package=ggpointdensity. (Accessed 10 April 2022).

Kuhn, M., Vaughan, D., Hvitfeldt, E., RStudio, 2022. Yardstick: Tidy Characterizations of model performance [WWW Document]. URL. https://CRAN.R-project.org/package=yardstick. (Accessed 10 April 2022). accessed.

LeDell, E., Poirier, S., 2020. H2o Automl: Scalable Automatic Machine Learning. In: Presented at the Proceedings of the AutoML Workshop at ICML.

Liashchynskyi, Petro, Liashchynskyi, Pavlo, 2019. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS arXiv preprint arXiv:1912.06059.

Mahjoobi, J., Adeli Mosabbeb, E., 2009. Prediction of significant wave height using regressive support vector machines. Ocean Eng. 36, 339–347. https://doi.org/10.1016/j.oceaneng.2009.01.001.

Meindl, E.A., Hamilton, G.D., 1992. Programs of the national data buoy center. Bull. Am. Meteorol. Soc. 73, 985–994. https://doi.org/10.1175/1520-0477(1992)073<0985:POTNDB>2.0.CO. ;2.

Montero, P., Vilar, J.A., 2015. TSclust: an R package for time series clustering. J. Stat. Software 62, 1–43. https://doi.org/10.18637/jss.v062.i01.

Panfilova, M.A., Kuznetsova, A.M., Titchenko, Yu.A., Sergeev, D.A., Troitskaya, Yu.I., Karaev, V.Yu., 2021. Methods of comparing the wave model simulation data with the KA-BAND radar data. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. Presented at the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 7537–7540. https://doi.org/10.1109/IGARSS47720.2021.9555041.

Pirhooshyaran, M., Snyder, L.V., 2020. Forecasting, hindcasting and feature selection of ocean waves via recurrent and sequence-to-sequence networks. Ocean Eng. 207, 107424 https://doi.org/10.1016/j.oceaneng.2020.107424.

Pokhrel, P., 2021. Machine Learning in Weakly Nonlinear Systems: A Case Study on Significant Wave Heights arXiv:2105.08583 [physics].

Quach, B., Glaser, Y., Stopa, J.E., Mouche, A.A., Sadowski, P., 2021. Deep learning for predicting significant wave height from synthetic aperture radar. IEEE Trans. Geosci. Rem. Sens. 59, 1859–1867. https://doi.org/10.1109/TGRS.2020.3003839.

Rana, M., Rahman, A., 2020. Multiple steps ahead solar photovoltaic power forecasting based on univariate machine learning models and data re-sampling. Sustain. Energy Grids Netw. 21, 100286 https://doi.org/10.1016/j.segan.2019.100286.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27, 379–423.

Shenoy, A.R., 2021. Grafify: an R Package for Easy Graphs, ANOVAs and Post-hoc Comparisons. https://doi.org/10.5281/ZENODO.5136508.

Shields, B.J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J.I.M., Janey, J.M., Adams, R.P., Doyle, A.G., 2021. Bayesian reaction optimization as a tool for chemical synthesis. Nature 590, 89–96.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. Adv. Neural Inf. Process. Syst. 25.

Storkey, D., Blockley, E.W., Furner, R., Guiavarc'h, C., Lea, D., Martin, M.J., Barciela, R. M., Hines, A., Hyder, P., Siddorn, J.R., 2010. Forecasting the ocean state using NEMO: the new FOAM system. J. Operat. Oceangraph. 3, 3–15.

Sun, A.Y., Scanlon, B.R., Save, H., Rateb, A., 2021. Reconstruction of GRACE total water storage through automated machine learning. Water Resour. Res. 57, e2020WR028666 https://doi.org/10.1029/2020WR028666.

Takaya, Y., Hirahara, S., Yasuda, T., Matsueda, S., Toyoda, T., Fujii, Y., Sugimoto, H., Matsukawa, C., Ishikawa, I., Mori, H., Nagasawa, R., Kubo, Y., Adachi, N., Yamanaka, G., Kuragano, T., Shimpo, A., Maeda, S., Ose, T., 2018. Japan Meteorological Agency/Meteorological Research Institute-Coupled Prediction System version 2 (JMA/MRI-CPS2): atmosphere–land–ocean–sea ice coupled prediction system for operational seasonal forecasting. Clim. Dynam. 50, 751–765. https://doi.org/10.1007/s00382-017-3638-5.

Tolman, H.L., 2009. User Manual and System Documentation of WAVEWATCH III TM Version 3.14. Technical Note. MMAB Contribution 276.

Truong, A., Walters, A., Goodsitt, J., Hines, K., Bruss, C.B., Farivar, R., 2019. Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. In: IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, pp. 1471–1479, 2019.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., Guyon, I., 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: analysis of the black-box optimization challenge 2020. In: NeurIPS 2020 Competition and Demonstration Track. PMLR, pp. 3–26.

Umesh, P.A., Behera, M.R., 2021. On the improvements in nearshore wave height predictions using nested SWAN-SWASH modelling in the eastern coastal waters of India. Ocean Eng. 236, 109550 https://doi.org/10.1016/j.oceaneng.2021.109550.

Vihtakari, M., 2022. ggOceanMaps: Plot Data on Oceanographic Maps Using Ggplot2. https://doi.org/10.5281/zenodo.4554714.

Wang, J., Song, Y., Liu, F., Hou, R., 2016. Analysis and application of forecasting models in wind power integration: a review of multi-step-ahead wind speed forecasting models. Renew. Sustain. Energy Rev. 60, 960–981. https://doi.org/10.1016/j.rser.2016.01.114.

Wilson, S., 2021. ParBayesianOptimization: parallel bayesian optimization of hyperparameters [WWW Document]. URL. https://CRAN.R-project.org/package=ParBayesianOptimization. (Accessed 25 April 2022).

Wu, J., Qin, L., Chen, N., Qian, C., Zheng, S., 2022. Investigation on a spring-integrated mechanical power take-off system for wave energy conversion purpose. Energy 245, 123318. https://doi.org/10.1016/j.energy.2022.123318.

Yang, H., Wang, H., Ma, Y., Xu, M., 2022. Prediction of wave energy flux in the Bohai sea through automated machine learning. J. Mar. Sci. Eng. 10, 1025. https://doi.org/10.3390/jmse10081025.

Yang, S., Deng, Z., Li, X., Zheng, C., Xi, L., Zhuang, J., Zhang, Zhenquan, Zhang, Zhiyou, 2021. A novel hybrid model based on STL decomposition and one-dimensional convolutional neural networks with positional encoding for significant wave height forecast. Renew. Energy 173, 531–543. https://doi.org/10.1016/j.renene.2021.04.010.

Zeng, Y., Zhang, J., 2020. A machine learning model for detecting invasive ductal carcinoma with Google Cloud AutoML Vision. Comput. Biol. Med. 122, 103861 https://doi.org/10.1016/j.compbiomed.2020.103861.

Zhang, H., Yan, J., Han, S., Li, L., Liu, Y., Infield, D., 2021. Uncertain accessibility estimation method for offshore wind farm based on multi-step probabilistic wave forecasting. IET Renew. Power Gener. 15, 2944–2955. https://doi.org/10.1049/rpg2.12227.