Electronic Research Archive

ERA, 32(7): 4543–4562.
DOI: 10.3934/era.2024206
Received: 01 December 2023
Revised: 20 May 2024
Accepted: 20 May 2024
Published: 23 July 2024

https://www.aimspress.com/journal/ERA

Research article

An improved stacking-based model for wave height prediction

Peng Lu[1,*], Yuze Chen[1], Ming Chen[1], Zhenhua Wang[1,3], Zongsheng Zheng[1], Teng Wang[2] and Ru Kong[2]

[1] College of Information Technology, Shanghai Ocean University, Shanghai 201306, China
[2] Shandong Provincial Institute of Land Space Data and Remote Sensing Technology, Shandong Ocean Bureau, Jinan 250002, China
[3] Fujian Provincial Key Laboratory of Coast and Island Management Technology Study, Fujian Institute of Oceanography, Xiamen 361013, China

* **Correspondence:** Email: plu@shou.edu.cn.

Abstract: Wave height prediction is hampered by the volatility and unpredictability of ocean data. Traditional single predictors are inadequate in capturing this complexity, and weighted fusion methods fail to consider inter-model correlations, resulting in suboptimal performance. To overcome these challenges, we presented an improved stacking-based model that combined the long short-term memory (LSTM) network with extremely randomized trees (ET) for wave height prediction. Initially, features with weak correlation to wave height were excluded using the Pearson correlation coefficient. Subsequently, a stacking ensemble tailored for time series cross-validation was deployed, employing LSTM and ET as base learners to capture temporal and feature-specific patterns, respectively. Lasso regression was utilized as the meta-learner, harmonizing these insights to improve accuracy by leveraging the strengths of each model across different dimensions of the data. Validation using datasets from four buoy stations demonstrated the superior predictive capability of our proposed model over single predictors such as temporal convolutional networks (TCN) and XGBoost, and fusion methods like LSTM-ET-BP.

Keywords: wave height prediction; stacking ensemble; time series split; LSTM network; extremely randomized trees

1. Introduction

As humanity expands its exploration of the oceans, accurate wave height prediction becomes increasingly vital for environmental monitoring, port development, and maritime navigation [1]. Scholars worldwide have extensively researched wave height prediction methods, including statistical models [2], numerical models [3] and machine learning. Statistical models, such as ARMA [4] and ARIMA [5], rely on predefined assumptions to extrapolate historical wave heights in sequence. However, these models assume time series data to be stationary and linear, a presumption that often does not align with the non-stationary and nonlinear nature of ocean waves, limiting their effectiveness in complex maritime environments [6]. Advancements in computing have enhanced the application of numerical models like WAM, SWAN and WAVEWATCH-III [7], which are based on mathematical equations to simulate physical phenomena. These models excel in broad oceanic regions [8], but their predictive accuracy decreases in the complex terrains of nearshore areas [9].

The rapid advancement of artificial intelligence has significantly advanced the use of machine learning in accurately predicting wave height [10]. For instance, artificial neural networks (ANN) have been effectively utilized for time-frequency wave height prediction, demonstrating superior performance compared to SWAN [11]. Similarly, the application of LSTM to wave height prediction has been investigated, revealing a superior performance relative to models such as ResNet and ELM [12]. While machine learning methods offer notable improvements over statistical and numerical models, the literature mentioned primarily focuses on the use of single predictors for wave height prediction. Given the high volatility and uncertainty associated with wave heights, relying on a single predictor is insufficient to fully explore the vast hypothesis space, thereby limiting the effective use of data. To address the limitations of single predictors, several studies have explored the integration of predictions from multiple models through weighted calculations. Arslan [13] utilized STL to decompose time series into seasonal, trend, and residual components, applying LSTM to fit the trend and residual components before merging these predictions with the seasonal component. The final predictions were then averaged with those from Prophet for the original data. Gungor et al. [14] employed BiLSTM, CNNLSTM, DCNN, DLSTM, and HDNN to predict remaining useful life (RUL), by formulating a mathematical optimization problem to determine the optimal weights for each model. However, these weighted fusion approaches, which linearly combine predictions from multiple models, often fail to capture the complex nonlinearity inherent in wave height. They also overlook crucial inter-model correlations, diminishing the effective utilization of model diversity. Additionally, the sensitivity of these methods to outliers can result in unstable predictions, thereby challenging their reliability for wave height prediction.

Based on the analysis provided, single models construct representations within a specific hypothesis space, while weighted fusion models merely combine multiple models through weights, neither utilizing the quantification of uncertainty in predictions. However, by assessing the predictive uncertainty of various models, one can analyze the correlation between their predictions. Consequently, this paper employs a stacking ensemble to evaluate the specific uncertainties of different models, thereby analyzing their interrelations and effectively amalgamating their predictive outcomes to reduce the overall predictive uncertainty. To this end, this paper proposes a wave height prediction model based on the improved stacking ensemble methodology, using LSTM and ET to model the temporal and feature dimension information of wave height data, respectively. The major contributions of this study are as follows:

i) Utilization of the Pearson correlation coefficient to determine the correlation between dataset features and wave height, eliminating redundant features and thereby enhancing operational efficiency and predictive accuracy.

ii) In the cross-validation process of stacking, we employ time series split than the traditional KFold method. This approach effectively prevents information leakage and maintains the chronological order of the data, thereby ensuring the cross-validation scenario that better aligns with practical applications.

iii) An information extraction module combining LSTM with ET is proposed. This module is designed to capture both the temporality of wave height and the correlations of features, extracting effective information from these dimensions. Additionally, Lasso regression is employed as the meta-learner within the stacking ensemble. It integrates the extracted information, providing a comprehensive perspective on data observation and enhancing the predictive accuracy.

2. Materials and methods

2.1. Relevant theories

2.1.1. Pearson correlation analysis

The Pearson correlation coefficient, a measure assessing the correlation between two variables, is frequently employed to filter out uncorrelated feature variables and reduce dimensionality [15]. Given two feature variables $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_i, \dots, y_n\}$, the formula of the Pearson correlation coefficient is calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \tag{1}$$

where $\bar{x}$ and $\bar{y}$ denote the mean values of the samples in the two feature variables X and Y respectively, while n represent the number of samples.

2.1.2. Long short-term memory network

Recurrent neural network (RNN) is extensively used in processing time-series data, storing previous input information within the network to influence current outputs [16]. However, RNN encounters challenges with long-term dependencies. As a variant of RNN, the long short-term memory (LSTM) network effectively addresses this issue [17], with its unit structure illustrated in Figure 1. LSTM mitigates the long-distance dependency problem caused by gradient vanishing, by introducing a gating mechanism that selectively adds or removes information during iterative propagation.

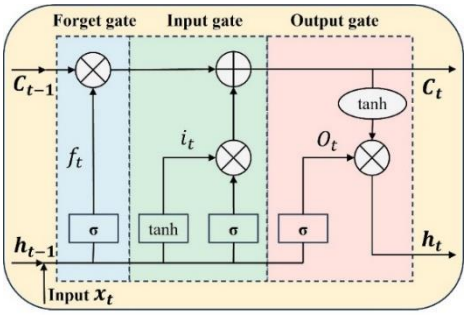


Figure 1. The unit structure of LSTM.

Given the input x_t at moment t, the unit state c_{t-1} and the unit output h_{t-1} at moment $t-1$, the working principle of the LSTM unit is as follows:

To begin with, the forgetting coefficient at moment t is calculated in the forget gate, which establishes the extent to which the cell state at moment $t-1$ is forgotten at moment t. The forgetting coefficient f_t is calculated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{2}$$

where W_f and b_f represent the weight and bias of the forget gate, respectively.

Furthermore, the input gate coefficient i_t and the candidate value vector $\widetilde{c_t}$ at moment t are computed in the input gate. The coefficient i_t dictates the extent to which the input x_t at moment t is retained. The formula is as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{3}$$

$$\widetilde{c_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{4}$$

where W_i and b_i denote the weight and bias of the input gate respectively. W_c and b_c denote the weight and bias of the unit state respectively.

After determining the forgotten and retained information, the cell state c_t at moment t is updated with the following formula:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \widetilde{c_t}, \tag{5}$$

where $f_t \cdot c_{t-1}$ specifies the information from the unit state c_{t-1} at moment $t-1$ that will be omitted, while $i_t \cdot \widetilde{c_t}$ defines the information to be incorporated into the unit state c_t at moment t.

Last, the output coefficient o_t is calculated in the output gate and the output h_t at moment t is determined. The output coefficient o_t and the output h_t are given respectively as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{6}$$

$$h_t = o_t \cdot tanh(c_t), \tag{7}$$

where W_o is the weight parameter of the output gate, and b_o is the bias associated with the output gate.

From the above principle, it can be seen that the unit output h_t is not solely derived from the input x_t at the current moment and the unit output h_{t-1} at the previous moment, but rather depends on the unit state c_t. The unit state c_t is controlled by a gating mechanism, which modulates the memory behavior. This mechanism selectively adds and removes information from c_t in each

iteration, as regulated by both the input and forget gates.

2.1.3. Extremely randomized trees

Extremely randomized trees, an evolution of random forests, employ a top-down method to generate a collection of decision trees [18]. The structure of extremely randomized trees is shown in Figure 2.

Compared with random forests, extremely randomized trees introduce greater randomness into the training process, which can reduce bias and variance more effectively. In extremely randomized trees, each decision tree is trained using the entire training set during the split process to minimize bias. Additionally, using randomly selected feature subsets and splitting thresholds aids in variance reduction [19]. After conducting numerous split tests on the feature subset and splitting thresholds, the node yielding the best score was selected for the next iteration [20]. This approach is repeated for each child node until a leaf node is reached. The final prediction is derived by aggregating the outputs of individual decision trees through mean calculation, thereby diminishing the model's sensitivity to noise [21].

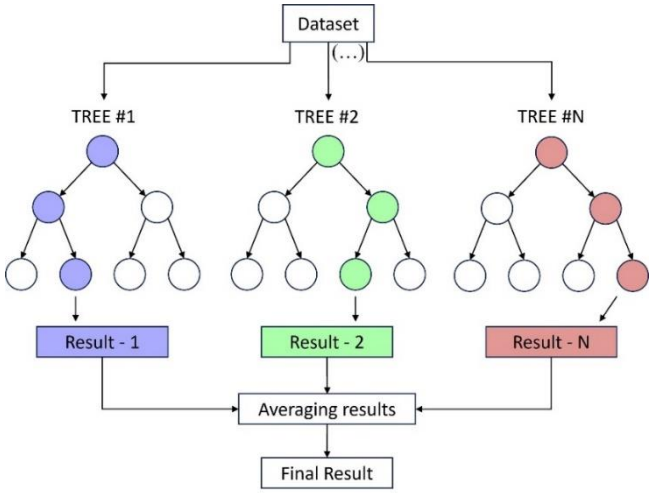


Figure 2. The structure of extremely randomized trees.

2.1.4. Time series cross-validation

Traditional cross-validation methods, such as KFold cross-validation, presuppose that samples are independent and identically distributed (i.i.d.). However, this assumption does not hold for time series data [22]. Applying KFold cross-validation to time series forecasting not only disrupts the temporal dependencies but also risks information leakage, leading to overfitting.

Given the abovementioned issues, KFold cross-validation is deemed unsuitable for time series forecasting. This paper adopts the time series split [23] method, tailored for time series analysis like wave height study. Time series split is a variation of KFold, and the 5-fold time series split process is depicted in Figure 3. Initially, the dataset is divided into training and test data. Subsequently, the training data is split into six equal-sized slices. The first slice forms the training set in the first fold, while the second becomes the validation set. In the second fold, the first two slices are combined to create the training set, with the third slice serving as the validation set. With each subsequent fold, the

training set is expanded by one slice, while the next slice in line serves as the validation set. This procedure is repeated until five distinct training-validation set pairs are formed over five iterations. Furthermore, this cross-validation procedure ensures that the indices for the training sets precede those of the validation sets in every iteration, maintaining the temporality of the series, thereby enabling the model to recognize the inherent trends within the data.

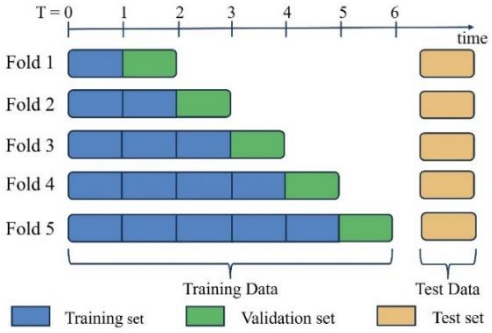


Figure 3. Time series split for k = 5.

2.2. Dataset construction

2.2.1. Study area

We utilize data from four stations in the Northeast Pacific Basin, the Gulf of Alaska, and the Sargasso Sea to empirically validate the efficacy of the proposed model. The geographic locations of these stations are depicted in Figure 4, and their detailed information is provided in Table 1, as sourced from the national data buoy center (NDBC).

Table 1. Details of the selected stations.

Station	Latitude	Longitude	Water Depth (m)	Period	Samples
41013	33.441N	77.764W	33	2022.03.01–2022.09.30	5136
46078	55.561N	152.599W	5361	2020.01.01–2020.07.31	5112
46084	56.614N	136.040W	1149	2021.01.01–2021.07.31	5088
46089	45.936N	125.793W	2375	2020.01.01–2020.07.31	5112

Figure 4. Positions of the selected stations.

2.2.2. Data preprocessing

In the process of data acquisition and transmission, occurrences of missing data are inevitable. Neglecting these missing values could result in significant information loss and disrupt the temporal continuity of the dataset. To address this, we employ an interpolation technique [24] to fill in missing values using the formula detailed below:

$$s_{t_j} = s_{t_i} + \frac{s_{t_k} - s_{t_i}}{t_k - t_i}(t_j - t_i), \tag{8}$$

where s_{t_j} denotes the filled data, t_j represents the moment at which the data is missing, t_i and t_k correspond to the previous and next moment of t_j, respectively.

Inputting datasets with diverse dimensions directly into the model may result in a bias towards features with larger magnitudes while diminishing the importance of those with smaller ones. We employ Min-Max normalization for standardization to convert data of different dimensions into dimensionless values, thereby ensuring that all features are on the same scale. The normalization formula is provided below:

$$x' = \frac{x - min(x)}{max(x) - min(x)}, \tag{9}$$

where x' represents the normalized value, x denotes the original value, and $\max(x)$ and $\min(x)$ are the maximum and minimum values of the dataset, respectively.

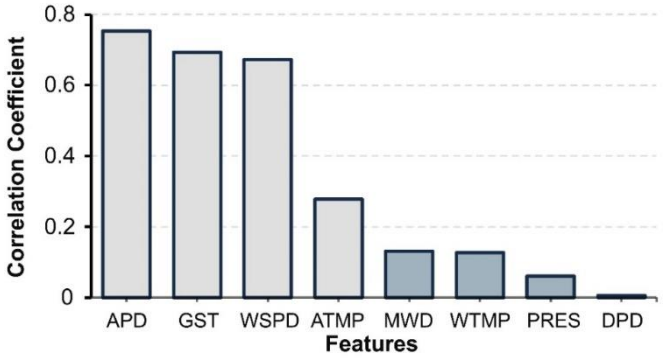


Figure 5. Pearson correlation coefficient for each feature.

Climate change is constantly affecting the marine environment, including ocean atmosphere circulation and water warming, and waves are the result of the interaction between the atmosphere and the ocean [25]. Thus, it is imperative to consider various factors such as winds, temperature, and wave periods that influence wave height. The dataset employed in this study encompasses eight influencing factors: average period (APD), wind gust (GST), wind speed (WSPD), air temperature (ATMP), mean wave direction (MWD), water temperature (WTMP), atmospheric pressure (PRES) and dominant wave period (DPD). However, using the entire feature data set in modeling can lead to irrelevant factors impacting the results and reducing accuracy. To mitigate this, it is crucial to evaluate the correlation between each feature and wave height to filter out factors that show little to no correlation.

This assessment is conducted using the Pearson correlation coefficient, and the results for each feature's correlation with wave height are displayed in Figure 5. The Pearson correlation coefficient ranging from 0 to 0.2 typically indicates a very weak or non-existent correlation. Hence, features with a correlation coefficient lower than 0.2 are excluded. Following the criteria, average period (APD), wind gust (GST), wind speed (WSPD), and air temperature (ATMP) are identified as influential predictive factors.

2.3. An improved stacking ensemble method combining LSTM and ET

Using a model to address the challenge where a single predictor fails to fully utilize available information and recognize the limitations of weighted fusion methods that neglect model intercorrelations, we propose a time series split-based stacking wave height prediction model, as depicted in Figure 6.

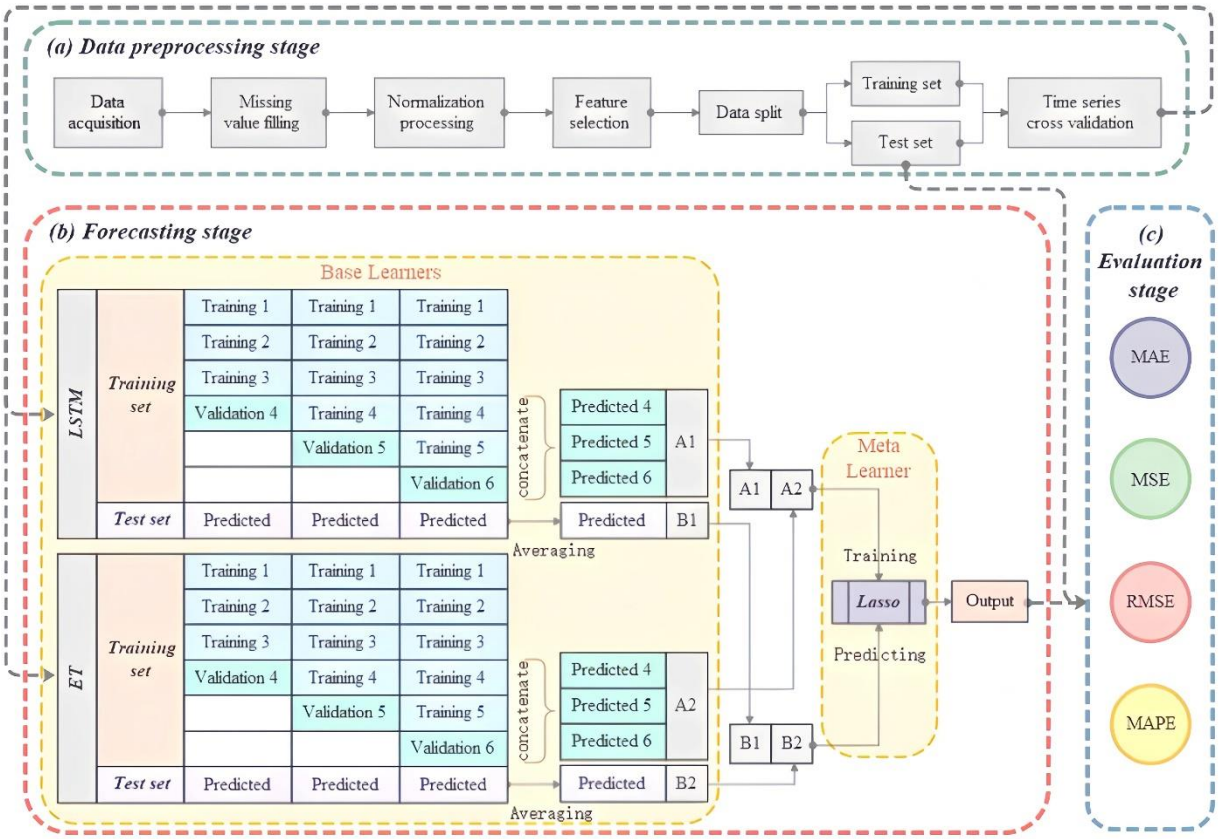


Figure 6. Stacking wave height prediction model based on time series split.

Stacking, as a model fusion technique, integrates multiple heterogeneous models through ensemble learning theory. This technique comprises base learners and a meta-learner. The base learners are tasked with extracting features from diverse perspectives, whereas the meta-learner specializes in generalizing and correcting errors in the predictions of the base learners. This collaboration significantly improves the overall performance. The stacking ensemble process is summarized in Algorithm 1.

Algorithm 1 Pseudo-code of **stacking ensemble**

Input:

N as the number of base models;

K as the number of splits in time series split;

$X_{train}^{(k)}$, $Y_{train}^{(k)}$ as the training data for the k^{th} split;

$X_{val}^{(k)}$, $Y_{val}^{(k)}$ as the validation data for the k^{th} split;

X_{test}, Y_{test} as the test data;

X_{meta_train}, Y_{meta_train} as the training data for the meta-learner;

X_{meta_test} as the features of test data for the meta-learner;

f_n as the n^{th} base learner; h as the meta-learner.

Output:

$\hat{Y}_{final}$.

for $n = 1 \rightarrow N$ **do**
  for $k = 1 \rightarrow K$ **do**
    Train f_n on $X_{train}^{(k)}$, $Y_{train}^{(k)}$;
    Predict $\hat{Y}_{val}^{(n,k)} = f_n\left(X_{val}^{(k)}\right)$;
    Predict $\hat{Y}_{test}^{(n,k)} = f_n(X_{test})$;
  end for
  Concatenate $\hat{Y}_{val}^{(n,1)}$, $\hat{Y}_{val}^{(n,2)}$, …, $\hat{Y}_{val}^{(n,k)}$ to form $X_{meta\ train}^{(n)}$;
  Concatenate $\hat{Y}_{val}^{(1)}$, $\hat{Y}_{val}^{(2)}$, …, $\hat{Y}_{val}^{(k)}$ to form Y_{meta_train};
  Aggregate $\hat{Y}_{test}^{(n,1)}$, $\hat{Y}_{test}^{(n,2)}$, …, $\hat{Y}_{test}^{(n,k)}$ to form $X_{meta\_test}^{(n)}$;
end for
Concatenate $\hat{X}_{meta\ train}^{(1)}$, $\hat{X}_{meta\ train}^{(2)}$, …, $\hat{X}_{meta\ train}^{(N)}$ to form X_{meta_train};
Concatenate $\hat{X}_{meta\ test}^{(1)}$, $\hat{X}_{meta\ test}^{(2)}$, …, $\hat{X}_{meta\ test}^{(N)}$ to form X_{meta_test};
Train h on X_{meta_train}, Y_{meta_train};
$\hat{Y}_{final} = h\left(X_{meta\_test}\right)$;

The choice of learners significantly influences the effectiveness of the predictive model. Adhering to the 'good but diverse' principle, we utilize LSTM and ET as base learners. These are selected for their ability to analyze data from temporal and feature-based perspectives. LSTM is a time series model that can handle long-term dependency, while ET is a tree-based model that comprehensively evaluates all possible feature divisions. We integrate lasso regression as the meta-learner after feature extraction by the base learners. Known for its robust generalizability, lasso regression systematically reduces model complexity by applying regularization terms that shrink the regression coefficients.

To reduce the risk of overfitting in the composite model, traditional stacking often utilizes KFold cross-validation for model training. However, applying KFold cross-validation to time series data may lead to information leakage and disrupt the inherent temporal correlation of the data. In light of this,

we adopt the time series split method as the cross-validation approach for the stacking process.

The stacking wave height prediction model, utilizing time series split for cross-validation, is executed in the following principal steps:

1) The dataset undergoes preprocessing, which includes filling in missing values, normalizing data, and filtering features.

2) For cross-validation, time series split is employed, where in the i^{th} iteration ($1 \leq i \leq$ k), the first i folds serve as the training set, and the $(i+1)^{th}$ fold as the validation set. Consistent with the literature [23], this study sets the k value in time series split to 5. Due to the relatively small training set compared to the validation set in the first two iterations, there is a potential risk of adversely affecting the cross-validation process [26]. Hence, the initial two iterations are excluded from this study. To maintain the monthly periodicity of wave height data and prevent the cross-validation process from disrupting it [27], the duration of each fold in time series split is set to one month.

3) During the time series cross-validation process, base learners are trained using the training set, producing predictions for both the validation and test sets. The structure of the temporal base learner (LSTM) is illustrated in Figure 7. In each cross-validation iteration, predictions for the validation set generated by the LSTM are vertically concatenated (denoted as A1), while predictions for the test set are averaged (denoted as B1).

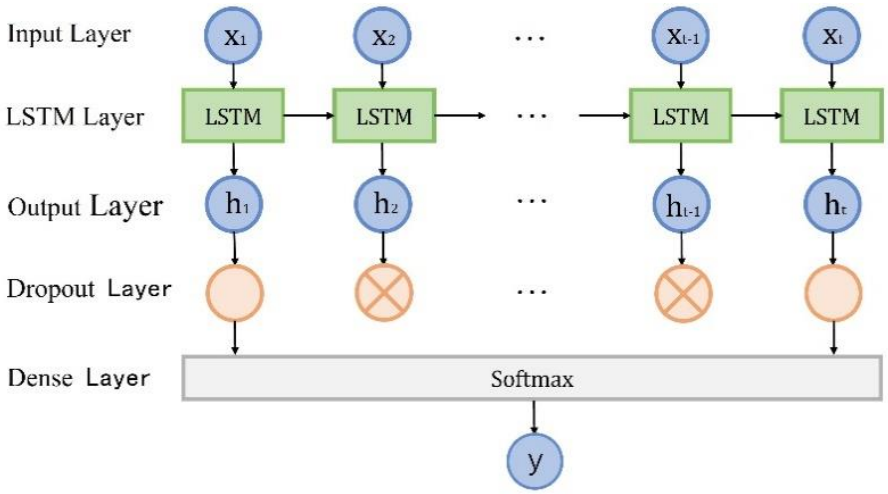


Figure 7. Model structure of temporal base learner LSTM.

Similarly, A2 and B2 can be obtained by the ET method. Since wave height prediction is a regression task, Eqs (10) and (11) are employed as scoring criteria for determining the splitting nodes in the feature-based learner. The Eqs (10) and (11) are shown below:

$$Score(s,S) = \frac{var\{y|S\} - \frac{|S_l|}{|S|}var\{y|S_l\} - \frac{|S_r|}{|S|}var\{y|S_r\}}{var\{y|S\}}, \tag{10}$$

$$var\{y|S\} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2, \tag{11}$$

where $Score(s,S)$ represents the scoring function, $var\{y|S\}$ denotes the variance of wave heights in sample S, S_l and S_r are the resultant left and right subsets of samples after the node split, n is

the total number of samples, y_i indicates the value of wave heights for the i^{th} sample, and $\bar{y}$ is the mean value of wave heights in S.

4) The training set for the meta-learner is created by horizontally concatenating A1 and A2, while B1 and B2 are combined in the same manner to form the test set for the meta-learner.

5) Employing the training set obtained from this horizontal concatenation, the meta-learner lasso is trained. The loss function, pivotal for the model's generalization capability, is delineated in the Eq (12). Following this, predictions are made using the horizontally concatenated test set. These predictions are then compared against the original data, with the predictive performance being quantified by established evaluation metrics.

$$\hat{\beta} = arg\min_{\beta}(\|y - X\beta\|^2 + \lambda\|\beta\|_1), \tag{12}$$

where β denotes the weight matrix, y represents the wave height series, X is the matrix of independent variables, and λ signifies the penalty factor.

2.4. Evaluation metrics

To quantify the predictive performance of the model, this paper employs mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) as evaluation metrics. The corresponding formulas are as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2, \tag{13}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|, \tag{14}$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}, \tag{16}$$

where n represents the number of samples, and y_t and $\hat{y}_t$ denote the true and predicted value of the sample at moment t, respectively.

3. Experimental results and comparative analysis

3.1. Selection of base and meta learners

The predictive efficiency of the stacking ensemble is significantly influenced by the performance of its constituent learners. It is crucial to carefully select the base learners and meta-learner to optimize the overall model, which we undertake through experimental analysis.

Table 2. Predictive performance of different base learner combinations.

Station	Algorithm	MAE	MSE	RMSE	MAPE
41013	LSTM-ET-Lasso-TSS	0.0742	0.0182	0.1347	0.0541
	LSTM-XGBoost-Lasso-TSS	0.0779	0.0205	0.1432	0.0554
	TCN-ET-Lasso-TSS	0.0767	0.0195	0.1395	0.0551
	TCN-XGBoost-Lasso-TSS	0.0841	0.0253	0.1591	0.0591
46078	LSTM-ET-Lasso-TSS	0.0711	0.0094	0.097	0.0502
	LSTM-XGBoost-Lasso-TSS	0.0768	0.0103	0.1014	0.054
	TCN-ET-Lasso-TSS	0.0793	0.0107	0.1033	0.0564
	TCN-XGBoost-Lasso-TSS	0.0754	0.0101	0.1006	0.0532
46084	LSTM-ET-Lasso-TSS	0.0685	0.0089	0.0943	0.0539
	LSTM-XGBoost-Lasso-TSS	0.0711	0.0093	0.0964	0.0561
	TCN-ET-Lasso-TSS	0.0743	0.0101	0.1005	0.0589
	TCN-XGBoost-Lasso-TSS	0.0727	0.0094	0.097	0.0585
46089	LSTM-ET-Lasso-TSS	0.0718	0.0098	0.099	0.048
	LSTM-XGBoost-Lasso-TSS	0.0819	0.0111	0.1052	0.057
	TCN-ET-Lasso-TSS	0.0768	0.0105	0.1024	0.052
	TCN-XGBoost-Lasso-TSS	0.088	0.012	0.1095	0.0627

Table 3. Predictive performance of different meta-learners.

Station	Algorithm	MAE	MSE	RMSE	MAPE
41013	LSTM-ET-Lasso-TSS	0.0742	0.0182	0.1347	0.0541
	LSTM-ET-LSTM-TSS	0.081	0.0268	0.1637	0.0543
	LSTM-ET-MLP-TSS	0.0771	0.0214	0.1463	0.0548
	LSTM-ET-SVR-TSS	0.0791	0.0225	0.15	0.056
46078	LSTM-ET-Lasso-TSS	0.0711	0.0094	0.097	0.0502
	LSTM-ET-LSTM-TSS	0.0733	0.0097	0.0985	0.0509
	LSTM-ET-MLP-TSS	0.0814	0.0113	0.1062	0.0564
	LSTM-ET-SVR-TSS	0.0768	0.0103	0.1015	0.0565
46084	LSTM-ET-Lasso-TSS	0.0685	0.0089	0.0943	0.0539
	LSTM-ET-LSTM-TSS	0.0742	0.0097	0.0987	0.0598
	LSTM-ET-MLP-TSS	0.07	0.009	0.095	0.0551
	LSTM-ET-SVR-TSS	0.0798	0.011	0.1048	0.0676
46089	LSTM-ET-Lasso-TSS	0.0718	0.0098	0.099	0.048
	LSTM-ET-LSTM-TSS	0.0765	0.0112	0.1058	0.0543
	LSTM-ET-MLP-TSS	0.0751	0.01	0.1002	0.0515
	LSTM-ET-SVR-TSS	0.0812	0.0109	0.1046	0.0617

For the base learners, we select LSTM and TCN as temporal base learner candidates, alongside XGBoost and ET for feature base learner candidates. As illustrated in Table 2, the LSTM-ET-Lasso-TSS combined model, utilizing LSTM for temporal and ET for feature analysis, demonstrates the most effective predictive performance. Theoretically, this superior performance can be attributed to the unique gating units of LSTM, which are adept at capturing long-term dependencies in time series data.

Moreover, the inherent randomness in ET and its exhaustive consideration of all feature splits play a crucial role in mitigating overfitting risks often encountered in multilayer model integration within a stacking ensemble.

For the meta-learner, we assessed lasso, LSTM, MLP and SVR, with results presented in Table 3. The analysis indicates that the LSTM-ET-Lasso-TSS model, utilizing lasso as the meta-learner, achieves the most favorable predictive results. Given the base learners' strong capabilities in information extraction, a weaker meta-learner like lasso is preferred over a stronger one. This choice mitigates the risk of overfitting and enables a more efficient combination of predictions from various base learners. Consequently, we select LSTM and ET as the base learners, with lasso as the meta-learner.

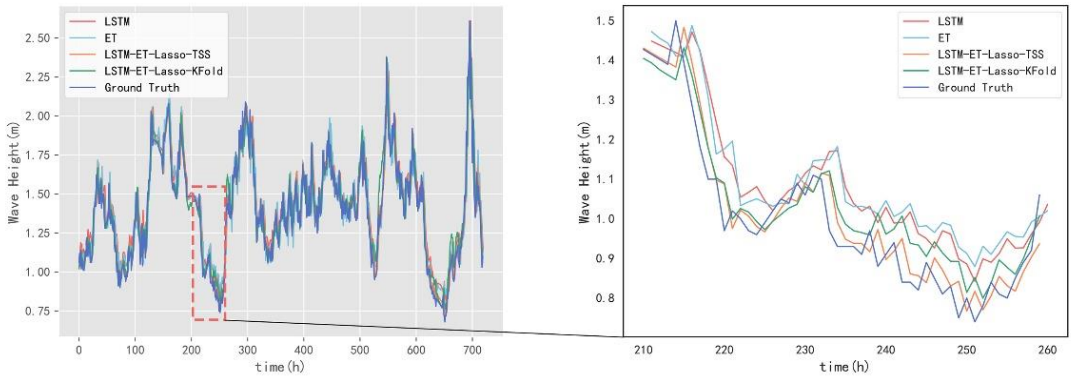
3.2. Ablation study

To validate the effectiveness of the proposed architecture, an ablation experiment was conducted. The critical model, LSTM-ET-Lasso-TSS, represents our proposed approach, employing time series split for cross-validation and integrating LSTM and ET via stacking. In contrast, LSTM-ET-Lasso-KFold uses traditional KFold for cross-validation, keeping all other aspects identical to LSTM-ET-Lasso-TSS. The predictive outcomes, as delineated in Table 4, reveal that compared to the LSTM and ET algorithms, the proposed model exhibits a reduced prediction error. This suggests that incorporating environmental factors and historical series into the prediction of wave heights significantly enhances forecasting performance. Furthermore, it demonstrates that the stacking ensemble effectively amalgamates these two distinct models. In addition, the proposed model outperforms the LSTM-ET-Lasso-KFold in evaluation metrics, implying that time series split more effectively preserves the 'temporal correlation' in time series data during cross-validation, thereby improving the model's overall predictive precision.

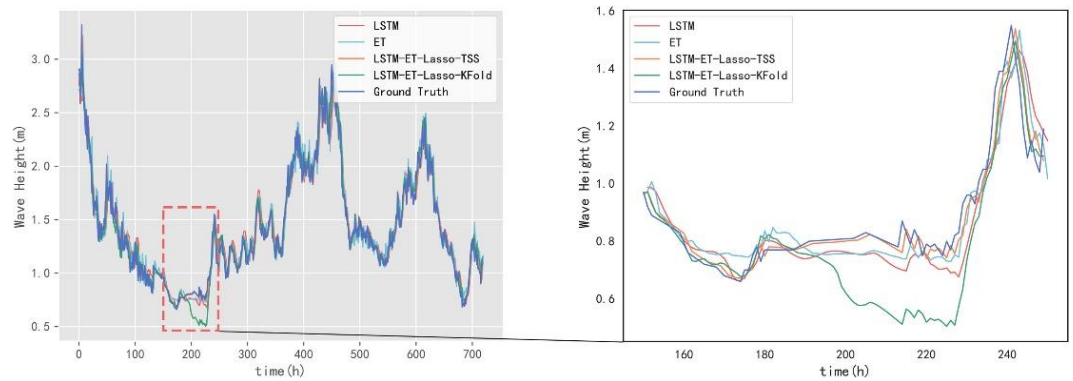
Table 4. Results of the ablation study.

Station	Algorithm	MAE	MSE	RMSE	MAPE
41013	LSTM-ET-Lasso-TSS	0.0742	0.0182	0.1347	0.0541
	LSTM-ET-Lasso-KFold	0.0808	0.0351	0.1874	0.0531
	LSTM	0.0876	0.0229	0.1514	0.072
	ET	0.0929	0.0634	0.2517	0.0592
46078	LSTM-ET-Lasso-TSS	0.0711	0.0094	0.097	0.0502
	LSTM-ET-Lasso-KFold	0.079	0.0108	0.1042	0.0559
	LSTM	0.0869	0.012	0.1091	0.0623
	ET	0.0857	0.0124	0.1115	0.0615
46084	LSTM-ET-Lasso-TSS	0.0685	0.0089	0.0943	0.0539
	LSTM-ET-Lasso-KFold	0.0706	0.0088	0.094	0.0559
	LSTM	0.0795	0.0106	0.1028	0.0627
	ET	0.0827	0.0131	0.1147	0.063
46089	LSTM-ET-Lasso-TSS	0.0718	0.0098	0.099	0.048
	LSTM-ET-Lasso-KFold	0.0827	0.013	0.114	0.0679
	LSTM	0.0811	0.0117	0.1083	0.0573
	ET	0.0863	0.0133	0.1152	0.059

To vividly illustrate the comparative effectiveness, this study selected data points within stations 46078 and 46089 where fluctuations significantly distinguish the performance of various models, as depicted in Figure 8. In the figure, the purple curve represents the actual series, while the orange curve denotes the predicted series of the proposed model. The green curve signifies the predicted series of the traditional stacking model implemented with KFold cross-validation, and the red and blue curves correspond to the predicted series of the LSTM and ET models, respectively. As evidenced in Figure 8, compared to the base learners LSTM and ET, as well as the traditional stacking model, the curve of the proposed model most closely aligns with the actual series, affirming the model's efficacy.



(a) Fitted curves for station 46078



(b) Fitted curves for station 46089

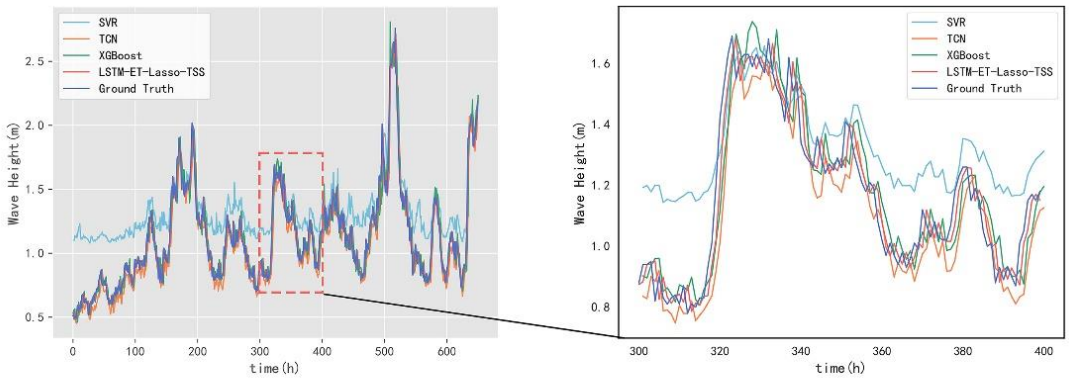
Figure 8. Fitted curves for stations 46078 and 46089.

3.3. Comparison study

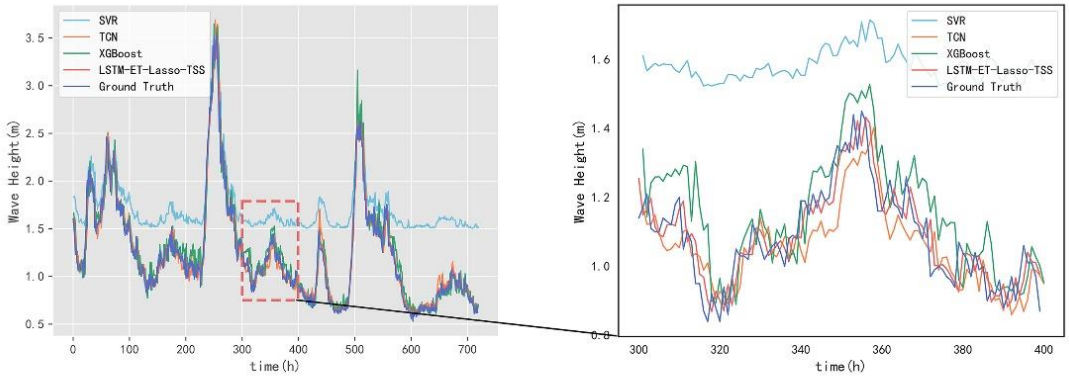
To assess the predictive accuracy and generalization capability of the proposed model, a comparative experiment was conducted against TCN, XGBoost and SVR models, using four data from distinct marine regions. Comparative outcomes are showcased in Table 5. The table demonstrates that the proposed model consistently exhibits the lowest values in evaluation metrics, indicating its superior predictive accuracy compared to the other models. Additionally, the results emphasize the model's strong generalization capabilities, further validating the effectiveness of the approach presented in this paper.

Table 5. Results of the comparative study.

Station	Algorithm	MAE	MSE	RMSE	MAPE
41013	LSTM-ET-Lasso-TSS	0.0742	0.0182	0.1347	0.0541
	TCN	0.0886	0.0236	0.1536	0.0744
	XGBoost	0.0917	0.0518	0.2276	0.0594
	SVR	0.2686	0.1929	0.4392	0.1917
46078	LSTM-ET-Lasso-TSS	0.0711	0.0094	0.097	0.0502
	TCN	0.0818	0.0113	0.1065	0.0586
	XGBoost	0.0961	0.0159	0.1261	0.0725
	SVR	0.8914	0.8796	0.9379	0.3905
46084	LSTM-ET-Lasso-TSS	0.0685	0.0089	0.0943	0.0539
	TCN	0.079	0.0109	0.1045	0.0636
	XGBoost	0.0909	0.0148	0.1218	0.0706
	SVR	0.4899	0.3023	0.5498	0.3045
46089	LSTM-ET-Lasso-TSS	0.0718	0.0098	0.099	0.048
	TCN	0.0849	0.012	0.1097	0.0669
	XGBoost	0.0821	0.012	0.1094	0.0566
	SVR	0.8709	0.9424	0.9708	0.3818



(a) Fitted curves for station 41013



(b) Fitted curves for station 46084

Figure 9. Fitted curves for stations 41013 and 46084.

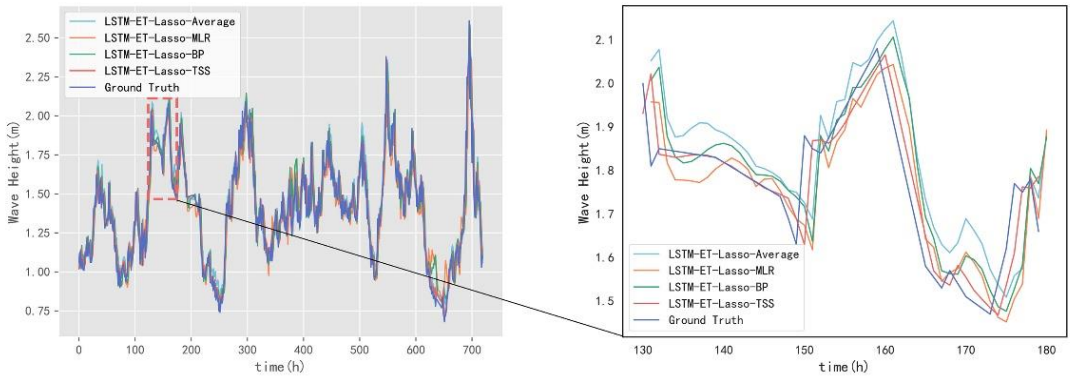
The fitting curves of various models at stations 41013 and 46084 are depicted in Figure 9. In this figure, the purple curve represents the actual series, the red curve indicates the predicted series of the proposed model, and the blue, orange, and green curves represent the predicted series of the classic wave height prediction models SVR, TCN, and XGBoost, respectively. It is evident from Figure 9 that the predictive curve of the proposed model more closely approximates the actual curve, particularly at the wave peaks (as illustrated by the times 375–385 in Figure 9(a) and 350–360 in Figure 9(b)), demonstrating a significant fitting advantage. Compared to SVR, the fitting results of TCN and XGBoost are superior, due to TCN's specially designed convolutional structure that captures the temporal dependencies, and XGBoost's ability to capture the impact of environmental factors on wave height through a combination of feature engineering and tree models. The predictive accuracy of the proposed model surpasses that of SVR, TCN, and XGBoost because it not only accounts for the temporal dependencies in wave height sequences but also considers the impact of environmental factors on wave height, effectively integrating these two considerations.

3.4. Study comparing with weighted fusion methods

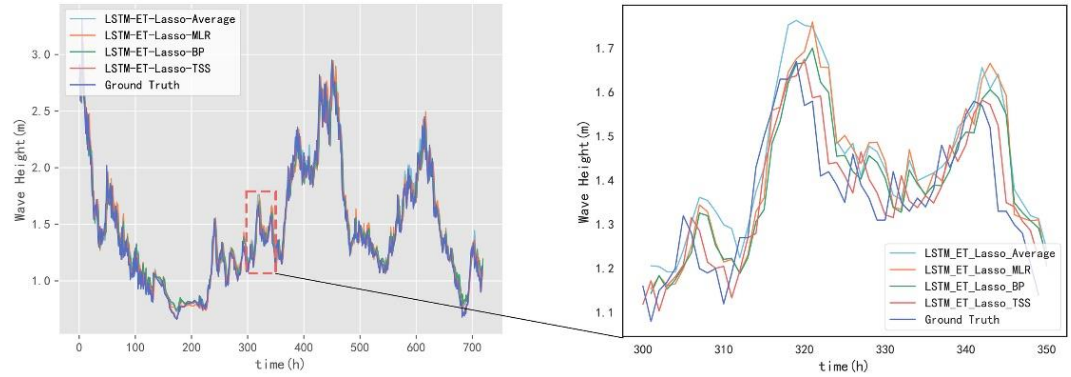
To assess the effectiveness of the improved stacking fusion technique, the proposed model was compared with various weighted fusion approaches combining LSTM and ET. The comparative results are shown in Table 6. In this comparison, LSTM-ET-Average integrates forecasts using an arithmetic mean, LSTM-ET-MLR employs MLR for weighted fusion, and LSTM-ET-BP utilizes BP for the same purpose. According to the results, the proposed model performs better than the weighted fusion approaches.

Table 6. Comparative results of fusion models.

Station	Algorithm	MAE	MSE	RMSE	MAPE
41013	LSTM-ET-Lasso-TSS	0.0742	0.0182	0.1347	0.0541
	LSTM-ET-Average	0.0853	0.0321	0.1792	0.0581
	LSTM-ET-MLR	0.0801	0.0327	0.1807	0.0543
	LSTM-ET-BP	0.0759	0.0315	0.1774	0.0514
46078	LSTM-ET-Lasso-TSS	0.0711	0.0094	0.097	0.0502
	LSTM-ET-Average	0.0788	0.0105	0.1024	0.0554
	LSTM-ET-MLR	0.0822	0.0122	0.1105	0.0602
	LSTM-ET-BP	0.0766	0.0105	0.1023	0.0544
46084	LSTM-ET-Lasso-TSS	0.0685	0.0089	0.0943	0.0539
	LSTM-ET-Average	0.071	0.0095	0.0976	0.055
	LSTM-ET-MLR	0.0761	0.0101	0.1005	0.0596
	LSTM-ET-BP	0.0733	0.0094	0.0967	0.0589
46089	LSTM-ET-Lasso-TSS	0.0718	0.0098	0.099	0.048
	LSTM-ET-Average	0.0802	0.0114	0.1069	0.0537
	LSTM-ET-MLR	0.0817	0.0116	0.1078	0.0553
	LSTM-ET-BP	0.0758	0.0101	0.1006	0.0525



(a) Fitted curves for station 46078



(b) Fitted curves for station 46089

Figure 10. Fitted curves for stations 46078 and 46089.

Figure 10 displays a comparison of the fitting curves between the proposed model and three weighted fusion methods. The purple curve denotes the actual series, the red curve represents the predicted series of the proposed model, and the blue, orange, and green curves symbolize the predictions of the three weighted fusion methods, respectively. It is evident from Figure 10 that, although the curves of the various models closely follow the trend of the actual series, the proposed model significantly outperforms the other three models in terms of fitting effectiveness, with the predictions of these three models being remarkably similar. This similarity is because the essence of these three models is the same, involving a straightforward weighted summation of the prediction results from LSTM and ET, without considering the correlation between the two base models. Conversely, the proposed model leverages stacking ensemble theory to effectively capture the nonlinear relationships between the base models, thereby enhancing the predictive accuracy.

In summary, the proposed model outperforms others in predictive accuracy and fitting precision. It suggests that the stacking fusion technique effectively captures inter-model correlations, thus efficiently integrating the diversity of various models.

4. Conclusions

We introduce an integrated prediction model utilizing an improved stacking ensemble, which effectively combines the LSTM network with the ET method to achieve accurate wave height

prediction. In contrast to single predictors, this model leverages the LSTM network for temporal information and the ET method for feature extraction, thereby enriching the model's observational diversity. In comparison to weighted fusion methods, the stacking ensemble adeptly captures inter-model nonlinearities, facilitating a more nuanced integration of correlations. Furthermore, by employing time series split for cross-validation within the stacking ensemble, than the traditional KFold approach, preserves the temporal correlation essential in time series data, thereby enhancing the model's ability to recognize trends.

To validate its effectiveness, the proposed model is compared with single predictors, weighted fusion models, and a traditional stacking fusion model, utilizing metrics such as MAE, MSE, MAPE and RMSE. Using station 41013 as an illustrative example, the MAEs for the TCN, XGBoost, SVR, LSTM-ET-Average, LSTM-ET-MLR, LSTM-ET-BP, and the traditional stacking model are 0.0886, 0.0917, 0.2686, 0.0853, 0.0801, 0.0759 and 0.0808 m, while the MAE for the proposed model stands at 0.0742 m. When compared directly, our model demonstrates a reduction in MAE by 0.0144, 0.0175, 0.1944, 0.0111, 0.0059, 0.0017, and 0.0066 m against the aforementioned models, corresponding to relative error reductions of 16.25, 19.08, 72.38, 13.01, 7.37, 2.24 and 8.17%, respectively. Evaluations conducted on four data from distinct marine stations confirm the proposed model's superior predictive performance, significantly improving wave height forecasting efficacy.

However, the model's integration of multiple base learners and the cross-validation process necessitates extensive training time. As new data emerges, requiring the model to be retrained, reducing the training duration presents a challenge. Therefore, future work will explore the adoption of incremental learning, enabling the model to update using only new data without the need to start training from scratch, thereby enhancing training efficiency. In addition, the prediction accuracy of the model will be further improved. Thus, how to ensure the prediction accuracy and training efficiency at the same time will be a focal point of future efforts.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the Research on Key Technology of Intelligent Extraction for Remote Sensing Monitoring of Marine Areas and Islands, grant number SDGP370000000202402001009A_001, the Shanghai Science and Technology Innovation Plan Project, grant number 20dz1203800, the Capacity Development for Local College Project, grant number 19050502100, the Fujian Provincial Public Welfare Project, grant number 2021R1007004, and the Study on a Deep Learning-Based Model for Extracting Coastlines of Islands, grant number FJCIMTS 2023-04.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. S. Shamshirband, A. Mosavi, T. Rabczuk, N. Nabipour, K. Chau, Prediction of significant wave height; comparison between nested grid numerical model, and machine learning models of artificial neural networks, extreme learning and support vector machines, *Eng. Appl. Comput. Fluid Mech.,* **14** (2020), 805–817. https://doi.org/10.1080/19942060.2020.1773932
2. A. Alqushaibi, S. J. Abdulkadir, H. M. Rais, Q. Al-Tash, M. G. Ragab, H. Alhussian, Enhanced weight-optimized recurrent neural networks based on sine cosine algorithm for wave height prediction, *J. Mar. Sci. Eng.,* **9** (2021), 524. https://doi.org/10.3390/jmse9050524
3. J. Swain, P. A. Umesh, P. K. Bhaskaran, A. N. Balchand, Simulation of nearshore waves using boundary conditions from WAM and WWIII–a case study, *ISH J. Hydraul. Eng.,* **27** (2021), 506–520. https://doi.org/10.1080/09715010.2019.1603087
4. P. T. D. Spanos, ARMA algorithms for ocean wave modeling, *J. Energy Resour. Technol.,* **105** (1983), 300–309. https://doi.org/10.1115/1.3230919
5. N. Zheng, H. Chai, Y. Ma, L. Chen, P. Chen, Hourly sea level height forecast based on GNSS-IR by using ARIMA model, *Int. J. Remote Sens.,* **43** (2022), 3387–3411. https://doi.org/10.1080/01431161.2022.2091965
6. W. Hao, X. Sun, C. Wang, H. Chen, L. Huang, A hybrid EMD-LSTM model for non-stationary wave prediction in offshore China, *Ocean Eng.,* **246** (2022), 110566. https://doi.org/10.1016/j.oceaneng.2022.110566
7. J. V. Björkqvist, O. Vähä-Piikkiö, V. Alari, A. Kuznetsova, L. Tuomi, WAM, SWAN and WAVEWATCH III in the Finnish archipelago—the effect of spectral performance on bulk wave parameters, *J. Oper. Oceanogr.,* **13** (2020), 55–70. https://doi.org/10.1080/1755876X.2019.1633236
8. M. Li, K. Liu, Probabilistic prediction of significant wave height using dynamic Bayesian network and information flow, *Water,* **12** (2020), 2075. https://doi.org/10.3390/w12082075
9. S. Contardo, R. Hoeke, P. Branson, V. Hernaman, T. Pitman, Hydrodynamic modelling for nearshore predictions, CSIRO, 2020.
10. S. Gracia, J. Olivito, J. Resano, B. Martin-del-Brio, M. de Alfonso, E. Álvarez, Improving accuracy on wave height estimation through machine learning techniques, *Ocean Eng.,* **236** (2021), 108699. https://doi.org/10.1016/j.oceaneng.2021.108699
11. E. Dakar, J. M. Fernández-Jaramillo, I. Gertman, R. Mayerle, R. Goldman, An artificial neural network based system for wave height prediction, *Coastal Eng. J.,* **65** (2023) 309–324. https://doi.org/10.1080/21664250.2023.2190002
12. S. Fan, N. Xiao, S. Dong, A novel model to predict significant wave height based on long short-term memory network, *Ocean Eng.,* **205** (2020), 107298. https://doi.org/10.1016/j.oceaneng.2020.107298
13. S. Arslan, A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data, *PeerJ Comput. Sci.,* **8** (2022), e1001. https://doi.org/10.7717/peerj-cs.1001
14. O. Gungor, T. S. Rosing, B. Aksanli, Opelrul: Optimally weighted ensemble learner for remaining useful life prediction, in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Detroit (Romulus), MI, USA, (2021), 1–8. https://doi.org/10.1109/icphm51084.2021.9486535

15. M. Murtaza, M. Sharif, M. Yasmin, M. Fayyaz, S. Kadry, M. Y. Lee, Clothes retrieval using M-AlexNet with mish function and feature selection using joint Shannon's entropy Pearson's correlation coefficient, *IEEE Access*, **10** (2022), 115469–115490. https://doi.org/10.1109/access.2022.3218322
16. F. Shahid, A. Zameer, M. Muneeb, Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM, *Chaos, Solitons Fractals*, **140** (2020), 110212. https://doi.org/10.1016/j.chaos.2020.110212
17. K. Cho, Y. Kim, Improving streamflow prediction in the WRF-Hydro model with LSTM networks, *J. Hydrol.*, **605** (2022), 127297. https://doi.org/10.1016/j.jhydrol.2021.127297
18. S. Heddam, Intelligent data analytics approaches for predicting dissolved oxygen concentration in river: Extremely randomized tree versus random forest, MLPNN and MLR, in *Intelligent Data Analytics for Decision-Support Systems in Hazard Mitigation*, (eds. R. C. Deo et al.), Springer, (2021), 89–107. https://doi.org/10.1007/978-981-15-5772-9_5
19. M. R. C. Acosta, S. Ahmed, C. E. Garcia, I. Koo, Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks, *IEEE Access*, **8** (2020), 19921–19933. https://doi.org/10.1109/access.2020.2968934
20. T. Yang, X. Liu, L. Wang, P. Bai, J. Li, Simulating hydropower discharge using multiple decision tree methods and a dynamical model merging technique, *J. Water Resour. Plann. Manage.*, **146** (2020), 04019072. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001146
21. M. Gong, J. Wang, Y. Bai, B. Li, L. Zhang, Heat load prediction of residential buildings based on discrete wavelet transform and tree-based ensemble learning, *J. Build. Eng.*, **32** (2020), 101455. https://doi.org/10.1016/j.jobe.2020.101455
22. A. D. Lainder, R. D. Wolfinger, Forecasting with gradient boosted trees: Augmentation, tuning, and cross-validation strategies: Winning solution to the M5 Uncertainty competition, *Int. J. Forecasting*, **38** (2022), 1426–1433. https://doi.org/10.1016/j.ijforecast.2021.12.003
23. M. Hasanov, M. Wolter, E. Glende, Time series data splitting for Short-Term load forecasting, in *PESS+ PELSS 2022; Power and Energy Student Summit*, Kassel, Germany, (2022), 1–6.
24. G. Huang, Missing data filling method based on linear interpolation and lightgbm, *J. Phys. Conf. Ser.*, **1754** (2021), 012187. https://doi.org/10.1088/1742-6596/1754/1/012187
25. B. G. Reguero, I. J. Losada, F. J. Méndez, A recent increase in global wave power as a consequence of oceanic warming, *Nat. Commun.*, **10** (2019), 205. https://doi.org/ 10.1038/s41467-018-08066-0
26. J. Korstanje, Model evaluation for forecasting, in *Advanced Forecasting with Python* (eds. J. Korstanje), Springer, (2021), 36–38. https://doi.org/10.1007/978-1-4842-7150-6_2
27. C. Xiao, F. He, Q. Shi, W. Liu, A. Tian, R. Guo, et al., Evidence for lunar tide effects in Earth's plasmasphere, *Nat. Phys.*, **19** (2023), 486–491. https://doi.org/10.1038/s41567-022-01882-8

AIMS Press