

Article

Prediction Model of Ammonia Nitrogen Concentration in Aquaculture Based on Improved AdaBoost and LSTM

Yiyang Wang¹, Dehao Xu², Xianpeng Li² and Wei Wang^{2,*} 

¹ School of Electrical and Automation Engineering, Liaoning Institute of Science and Technology, Benxi 117004, China; yaopeng17@sohu.com

² College of Information Engineering, Dalian Ocean University, Dalian 116023, China; xudehaopro@sina.com (D.X.); 15829169220@163.com (X.L.)

* Correspondence: ww_wangwei@dlou.edu.cn

Abstract: The concentration of ammonia nitrogen is significant for intensive aquaculture, and if the concentration of ammonia nitrogen is too high, it will seriously affect the survival state of aquaculture. Therefore, prediction and control of the ammonia nitrogen concentration in advance is essential. This paper proposed a combined model based on X Adaptive Boosting (XAdaBoost) and the Long Short-Term Memory neural network (LSTM) to predict ammonia nitrogen concentration in mariculture. Firstly, the weight assignment strategy was improved, and the number of correction iterations was introduced to retard the shortcomings of data error accumulation caused by the AdaBoost basic algorithm. Then, the XAdaBoost algorithm generated and combined several LSTM su-models to predict the ammonia nitrogen concentration. Finally, there were two experiments conducted to verify the effectiveness of the proposed prediction model. In the ammonia nitrogen concentration prediction experiment, compared with the LSTM and other comparison models, the RMSE of the XAdaBoost–LSTM model was reduced by about 0.89–2.82%, the MAE was reduced by about 0.72–2.47%, and the MAPE was reduced by about 8.69–18.39%. In the model stability experiment, the RMSE, MAE, and MAPE of the XAdaBoost–LSTM model decreased by about 1–1.5%, 0.7–1.7%, and 7–14%. From these two experiments, the evaluation indexes of the XAdaBoost–LSTM model were superior to the comparison models, which proves that the model has good prediction accuracy and stability and lays a foundation for monitoring and regulating the change of ammonia nitrogen concentration in the future.



Citation: Wang, Y.; Xu, D.; Li, X.; Wang, W. Prediction Model of Ammonia Nitrogen Concentration in Aquaculture Based on Improved AdaBoost and LSTM. *Mathematics* **2024**, *12*, 627. <https://doi.org/10.3390/math12050627>

Received: 17 January 2024

Revised: 16 February 2024

Accepted: 19 February 2024

Published: 20 February 2024



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

Keywords: aquaculture; adaptive boosting algorithm; LSTM; combined prediction

MSC: 37N99

1. Introduction

In the world, China is the first aquaculture country and the first fishery country [1], and the output of aquatic products has ranked first in the world for 29 consecutive years since 1989 [2]. China's fish production from aquaculture has been far more than other countries since 1991. Aquaculture has changed the status quo of traditional capture fisheries, and aquaculture production has exceeded capture production. In the aquaculture process, the complex culture environment, high culture density, biological excretion, and other reasons will lead to the rise of ammonia nitrogen concentration in the water body. The increase in ammonia nitrogen concentration will lead to the rise of toxicity in water, which will lead to the poisoning of large areas of aquatic animals. If water quality control is not carried out in time, it will lead to many deaths [3]. Therefore, in aquaculture, it is crucial to monitor and control the ammonia concentration in advance.

At present, the detection of ammonia nitrogen concentration in China is divided into two categories [4]. One is the sampling laboratory detection method, which is highly accurate, but the time of detection is long, the cost is high, and the experimental results

cannot be reproduced. The other is the reagent method, using reagents or test paper on-site inspection; the method is short in time, but the accuracy is low, and both detection methods cannot provide a stable basis for the current water quality control. With the development of deep learning, the emergence of the neural network prediction model provides a new method for monitoring and controlling water quality in aquaculture. In the past, many scholars used mathematical models or statistical models to predict water quality parameters in aquaculture water quality; the standard models mainly included linear regression [5] and multiple regression models [6]. The researchers used these models to establish a linear relationship between water quality parameters and input variables. However, due to the complex environment of aquaculture water quality and the coupling relationship between water quality parameters, although these regression models can realize the prediction of water quality parameters, the prediction accuracy is difficult to guarantee. With deep research, water quality parameter prediction models gradually considered the nonlinear relationship between input and output. Commonly used models include back propagation neural network (BPNN) and support vector machine (SVM) neural network. Chen et al. [7] used backward propagation neural network (BPNN), the adaptive neural fuzzy inference system (ANFIS) method, and multiple linear regression (MLR) models to predict the dissolved oxygen concentration of Feitsui Reservoir in northern Taiwan. The results showed that the BPNN model and MLR model are less accurate than the ANFIS. Nong et al. [8] used an SVM neural network model coupled with data denoising, feature selection technology, and parameter optimization methods to establish a dissolved oxygen model. They used the model to predict dissolved oxygen in different locations of the South-to-North Water Transfer project. The results showed that the SVM neural network coupled with multiple intelligent technologies is more accurate than the comparison model. Liu Ru et al. [9] used the Pearson correlation coefficient (PCC) to analyze the correlation of each index in water and monitor water quality with five indicators with high correlation. The K nearest neighbors (KNN) algorithm was used. The integrated learning AdaBoost algorithm and decision tree three machine learning algorithms predicted the monthly average value of ammonia nitrogen concentration in a water body. This method provided solutions for water body data analysis and adopted a variety of algorithms to verify the prediction. The structure of these models is relatively simple and cannot be predicted with high precision. Recurrent neural networks (RNN) have emerged with the continuous development of theories and techniques. RNN can retain the history information in the data by hiding the state and contributing the included history information to the calculation of the current time step. In addition, RNN can capture historical patterns in the data as it is input, improving the accuracy of the final prediction. Sagar et al. [10] used wireless sensors to collect pollutant concentration data in Indian cities and input the acquired data into the RNN model. Experiments showed that the RNN model performed well on different urban pollutant data sets. However, RNN has the phenomenon of gradient disappearance or explosion, so many scholars have improved the RNN to obtain two models: Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU). Nitzan et al. [11] combined climate measurement data with water quality data and used LSTM to predict ammonia nitrogen and nitrate concentrations in water after wastewater treatment. The final experiment showed that the accuracy of ammonia nitrogen concentration was 99%, and the accuracy of nitrate concentration was 90%. Huan et al. [12] used a gradient lifting decision tree to select data features and LSTM to predict dissolved oxygen; the results showed that the accuracy was better than that of the comparison model PSO-LSSVM. In data acquisition, the sensor will cause noise in the collected data due to human or environmental factors. Therefore, some scholars have combined the denoising intelligence algorithm with the neural network model and applied it to water quality parameter prediction (see Yan et al. [13]). First, the original ammonia nitrogen concentration data is divided into several sub-sequences using variational modes. Secondly, the GRU model is used to model and predict the subsequence. Finally, the prediction results of the subsequences are added together. The results show that the prediction accuracy is improved compared with

the comparison model. In addition, some scholars use intelligent algorithms to enhance the performance of neural network models. Jannatul et al. [14] used the particle swarm optimization (PSO) method to optimize the hyperparameters in the LSTM model to improve the LSTM model’s ability to learn time series features. They conducted experiments using the water level observation data of observation stations along the Brahmaputra River, Ganges River, and Megna River in Bangladesh. The final results showed that the PSO–LSTM model was superior to the ANN, PSO–ANN, and LSTM models in predicting water level. Ganiyu et al. [15] adopted adaptive boosting (AdaBoost) to improve the LSTM and GRU models and conducted experiments on the crude oil price data set. By comparing single LSTM and GRU models, it has been proved that AdaBoost improves the prediction performance of the LSTM and GRU models.

This paper proposes a solution to the issue of inaccurate and unstable prediction models for ammonia nitrogen concentration. The proposed solution is a combined XAdaBoost and LSTM prediction model that utilizes turbid culture data collected by the intensive seawater circulation control system. The model takes inputs such as temperature, dissolved oxygen, pH, and conductivity and predicts the ammonia nitrogen concentration for the next moment. To address the error accumulation problem of the AdaBoost base algorithm, the paper proposes a new weight assignment strategy, introduces a corrective iteration number, and combines the improved AdaBoost algorithm with multiple LSTM sub-models for prediction.

2. Introduction of Basic Theory

2.1. Adaptive Enhancement Algorithm

Adaptive boosting [16] (AdaBoost) is a common type of integrated learning algorithm boosting class, first applied to classification problems and gradually used in regression tasks as the algorithm evolves [17]. AdaBoost’s key feature is its adaptability. This is achieved by adjusting the weights of data points based on their error rates in the previous sub-prediction model. Data with high error rates are given more weight, while data with low error rates are given less weight. This process is repeated for each sub-prediction model, using the newly weighted data to train the next model. Each sub-prediction model builds on the previous one in the iteration process, aiming to solve the data poorly processed by the earlier sub-model. When the number of iterations or the error rate satisfies the set value, the algorithm will stop the operation and save the final model. The principle of AdaBoost is shown in Figure 1.

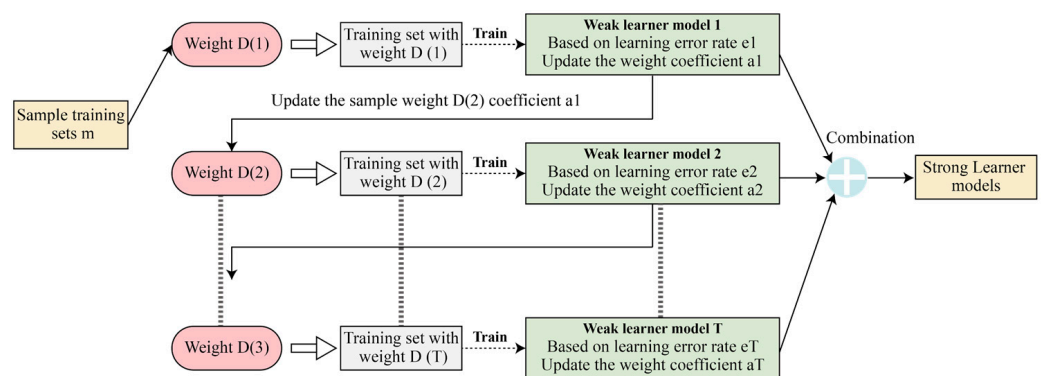


Figure 1. Schematic diagram of adaptive enhancement algorithm.

As can be seen from Figure 1, the algorithm can be divided into three modules: the initial value weight assignment module, the sub-prediction model training module, and the sub-model combination module. The initial value weight assignment module mainly assigns initial values to the data sample weights $D(1)$. m sample training sets are shown in the figure, and then each data initial value is $1/m$. Suppose our training samples are:

$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots (x_m, y_m)\} \tag{1}$$

where represents the sample set, x_m, y_m represents the input quantity and output quantity, respectively.

Then the output weights of the n th weak learner in the training set are

$$D(n) = (w_{n1}, w_{n2}, w_{n3}, w_{n4}, \dots, w_{nm}); w_{1i} = \frac{1}{m}; i = 1, 2, 3, \dots, m \quad (2)$$

The sub-model training module refers to training the weak learner model to the optimal state by selecting part of the data set, comparing the predicted values with the actual values in the training process, obtaining the training data errors, and increasing the data weights with significant error rates; otherwise, reducing them. The updated weights are applied to the next sub-model in the prediction process, and each sub-model puts more operations on the data with more significant error rates based on the previous sub-model. The sub-model combination is the process of combining trained models to form a robust learning model, in which more weights are assigned to the sub-prediction models with small error rates to ensure the accuracy of the vital learning model.

2.2. Long and Short-Term Memory Neural Networks

Recurrent neural networks can handle time series problems, but with the continuous input of time series, the traditional recurrent neural networks are prone to gradient disappearance or gradient explosion due to the abnormal computation of gradients, which leads to the degradation of model accuracy [18]. To solve the impact caused by gradient disappearance or gradient explosion of recurrent neural networks, the LSTM model is generated by introducing three gating units and memory units based on the improvement of traditional recurrent neural networks [19]. The control of the information replaces the information retained in the memory unit through the three gating units; the principle of the LSTM algorithm is shown in Figure 2.

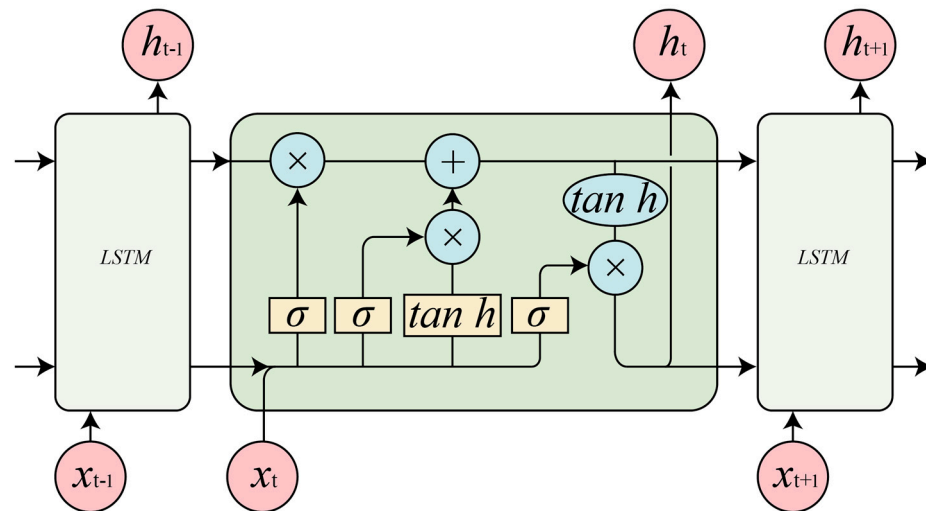


Figure 2. LSTM algorithm schematic.

Figure 2 shows that the input of each LSTM cell contains the current moment input information and the previous moment memory information, and the LSTM cell contains forgetting gates, input gates, and output gates to filter the information in the memory cell by forgetting gates:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (3)$$

where, f_t is the output activation value, σ is the Sigmoid function, after Sigmoid can obtain the number between 0 and 1, it will be multiplied with the memory unit of the previous moment bit by bit for memory retention and forgetting, W_f represents the weight matrix in the forget gate, b_f represents the bias matrix in the forget gate, h_{t-1} represents the output

state at the previous time, x_t represents the input at the current time. The final result is a number between 0 and 1. When f_t is zero, the information is forgotten, and when f_t is 1, the information is retained.

The input gate determines how much information about the current moment is retained. The purpose of the input gate is to determine the importance of the current input to the overall situation:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (5)$$

where i is similar to the formula of the forgetting gate, which filters the input at the current moment. The candidate output state of the current moment of the input gate is a nonlinear transformation of the hidden state of the previous moment and the input of the current moment. \tanh is another activation function, no matter how large the input of the activation function is; the final output is the number in the interval $[-1,1]$. W represents the weight matrix in the output gate, and C represents the candidate state weight matrix in the output gate. Both W and C are bias matrices.

The forgetting gate and the input gate work together to update the memory, and the new memory contains the memory retained by the forgetting gate and the memory added by the input gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

C_t represents the state of the cell at the current time; it can be seen from Equation (6) that the current cell state is affected by the forgetting gate as well as the input gate.

Filter which of the current memory cells are used as the hidden state and output for the next moment through the output gate.

$$O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (7)$$

O_t is the output value of the output gate at the current time and W_o is the weight matrix of the output gate at the current time. b_o is the bias matrix of the output gate. The output state at the current moment is calculated using the following formula:

$$h_t = O_t * \tanh(C_t) \quad (8)$$

h_t will be fed into the next LSTM cell as the input information at the next moment.

LSTM improves the long-term dependence problem in RNN by introducing the structure of the cell state including the forgetting gate, input gate, and output gate, and its performance is usually better than the temporal recurrent neural network and hidden Markov model. LSTM itself can also be used as a complex nonlinear unit to construct a larger deep network and this sets the stage for our following research.

3. XAdaBoost-LSTM Based Ammonia Concentration Prediction Model

3.1. Fundamentals of Predictive Models

The LSTM ammonia nitrogen concentration prediction model is based on time series prediction, taking into account the effects of temperature, salinity, and pH while also considering the temporal nature of ammonia nitrogen concentration. However, the stability of a single LSTM ammonia nitrogen concentration prediction model is poor, so ensemble learning algorithms are selected to combine multiple sub models to improve model accuracy and robustness. The proposed model mainly combines the advantages of the XAdaBoost algorithm and LSTM model. The basic idea of XAdaBoost-LSTM is to use LSTM as the base model and enhance it using the XAdaBoost algorithm. Specifically, multiple LSTM models can be trained, each using different data sets and features, and their prediction results can be combined to form a more accurate and robust model. The basic working principle of the XAdaBoost LSTM model is as follows: Firstly, the weight W is initialized through the total number of experimental samples, and then the LSTM model is used as a sub model. The

LSTM sub model is trained using the sample data with weight W to obtain the prediction results. The weight of the sample data is recalculated based on the prediction results, and the weight of data with high prediction error rates is amplified; the sample weight of data with small prediction error coefficients is reduced. When the number of predicted sample points with an error rate greater than the set error rate is equal to T , the sample weight is set to 0. Linear combination is performed on weak learners based on their performance, and the proportion of each sub model in the strong learning model is determined based on the performance of each sub model. Finally, the final weight is combined with the model to obtain the final prediction model.

3.2. Improved Adaptive Enhancement Algorithm (XAaBoost)

Prediction tasks can have error data during data collection and recording, and since most tasks are used to predict outliers, the sample error data cannot be accurately removed during preprocessing. Therefore, it is essential to identify the error data when AdaBoost is reassigned to the sample data. In this paper, we deal with the prediction task and propose that the XAdaBoost algorithm is used to improve the model prediction accuracy and robustness by combining previous research on classification tasks to address this drawback of AdaBoost's error accumulation.

Suppose the number of iterations of the AdaBoost algorithm is T . Given an error range according to the model, it is divided into four categories according to the number of accurate predictions.

- (1) If the prediction error is less than d in each of the T iterations, then the prediction value of each iteration is less than the error range, which means that each sub-model is correct in processing this sample.
- (2) The number of errors less than d in T iterations is greater than $T/2$, which means that the number of correct predictions in T sub-models is greater than the number of errors in this data.
- (3) Similarly, less than $T/2$ means that the number of correctly predicted sub-models is less than the number of incorrect ones.
- (4) The error rate is greater than d in all T iterations, which means that none of the samples are accurate after T iterations.

Based on the above four cases, one more correction iteration is performed, at which point the sample weight update strategy is as follows:

The data in (4) above are considered as wrong data and the weights are assigned to 0; case (3) is assigned a larger value of weights than case (4), and since there are less than the number of correct ones, the weights are low in the previous sub-models and are assigned more in the last time for correction. Then, as per (1), each prediction results in lower correct weights. Therefore, $T + 1$ iterations are used, and the error data are removed in the last iteration, and the weights are assigned more reasonably; the last sub-model is used to correct the previous model containing error data. The previous sub-models are combined on the basis of the last correction.

3.3. Constructing a Prediction Model

In this paper, we used XAdaBoost combined with LSTM, which could effectively take into account the data temporality as the ammonia nitrogen concentration prediction sub-model, and used the AdaBoost integrated learning algorithm to iteratively enhance the LSTM sub-model.

(1) Assume that the number of iterations $t = 1, 2, \dots, T$ and the initialized data weights are

$$W_t(i) = 1/N \quad (9)$$

where $W_{T-1}(i)$ denotes the sample weight of the algorithm at the T th iteration of the LSTM sub-model, and N denotes the total number of samples of the experiment.

(2) The LSTM was used to build the sub-model $W_{T-1}(i)$ to train the sample data with the weight of $W_{T-1}(i)$ to obtain the prediction result \hat{y} and save the obtained sub-model.

(3) The sample data weights were recalculated by the prediction results \hat{y} , and the weights of the data with large prediction error rate were enlarged and trained by sub-model iterations.

$$e_{max}(x) = \max(y - \hat{y}) \tag{10}$$

$$e_{T-1}(x) = \frac{(y - \hat{y})}{e_{max}} \tag{11}$$

where: e_{max} denotes the maximum error, and e_{T-1} denotes the error rate of the sub-model at $T - 1$ iterations.

(4) After repeating iterations T times.

At the $T + 1$ st iteration, given the error rate d , $e_T(i)$ denotes the error of the i th sample in t iterations. If the number of $e_T(i) > d$ is equal to T let $W_T(i) = 0$.

(5) The weak learners were combined linearly according to their performance, and their weight in the strong learning model was judged by the performance of each sub-model:

$$\partial_T = \frac{1 - e_T}{e_T} \tag{12}$$

where, ∂_T denotes the weight of the T th sub-model to the strong learning model.

(6) After each training session, the training data weights could be updated by the data error rate and the model share:

$$W_{T+1} = \frac{W_T \partial_T^{1-e_T}}{Z_T} \tag{13}$$

$$Z_T = \sum_{i=1}^N W_T \partial_T^{1-e_T} \tag{14}$$

where Z_T denotes the normalization factor, in order to normalize the data weights after the iteration.

(7) The strong learning model was obtained by the combination of the final weights and the model.

$$H(x) = \sum_{t=1}^T \partial_t M_t(x) \tag{15}$$

where the output result of the final combined model represented the output of each sub-model, which represented the weight of the T -th sub-model. The above improved AdaBoost combination was used to generate the LSTM sub-model; the model structure is shown in Figure 3.

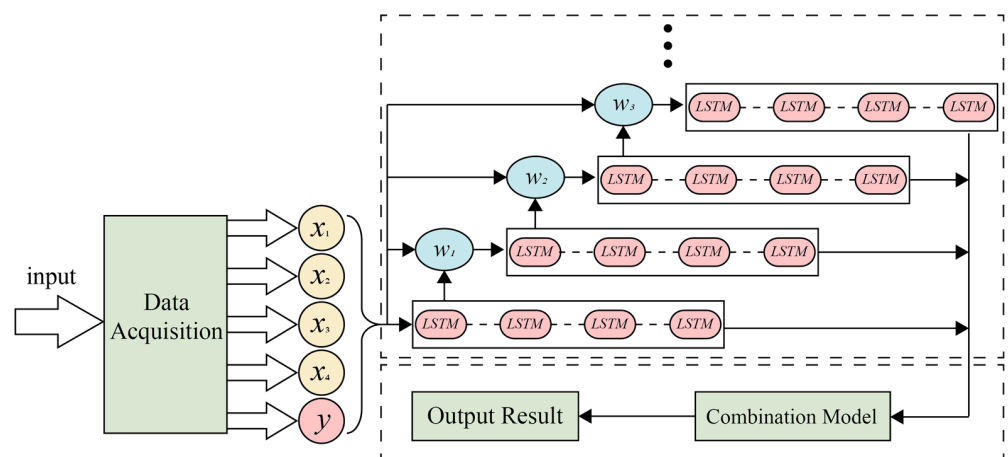


Figure 3. XAdaBoost-LSTM model frame diagram.

In Figure 3, We can divide the diagram into four parts: data input, sub-model training, sub-model combination, and result output. The data input part was mainly used to preprocess the data and specify the input and output variables for the preprocessed data. The sub-model training part was mainly used for data to train the XAdaBoost and LSTM models and update and determine model weights. The sub-model combination part was used to obtain the strong learning model by combining the final weights and the model and then outputting the final result. Such a model architecture inevitably has some defects. The model architecture contained two algorithms, XAdaBoost and LSTM, so the overall combined model also had problems, such as the model accuracy being affected by the balance of data set division and long training time.

3.4. Forecasting Process

The flow chart is divided into four parts: data acquisition, data pre-processing, model training and combination, and output. The data acquisition test object was turbot, which was reared in intensive recirculating seawater culture mode. The temperature, dissolved oxygen, pH, and conductivity data in the culture water were collected by sensors and saved to the corresponding database through PLC (programmable logic controller). A programmable controller was mainly used in the experiment to automatically realize the water level regulation, oxygen pump work, and sensor acquisition of the circulation aquaculture system. The nano reagent method [20] was used to measure ammonia nitrogen concentration, which was simple, rapid, and sensitive to meet the data collection requirements. The data pre-processing part consisted of three primary operations, namely: outlier processing, normalization, and partitioning of the training data set. If the original data contained outliers then the outliers were cleaned up and supplemented by linear interpolation. Since the original data had different units of each feature variable, the actual data was normalized in order to eliminate the effect of dimensionality [21], and was used to divide the data into training set and test set. The model training and combination part mainly used the training data to train the XAdaBoost–LSTM model and saved the final parameters to combine into a robust learner model. The output section outputted the prediction results of the training and test sets; the model prediction process is shown in Figure 4.

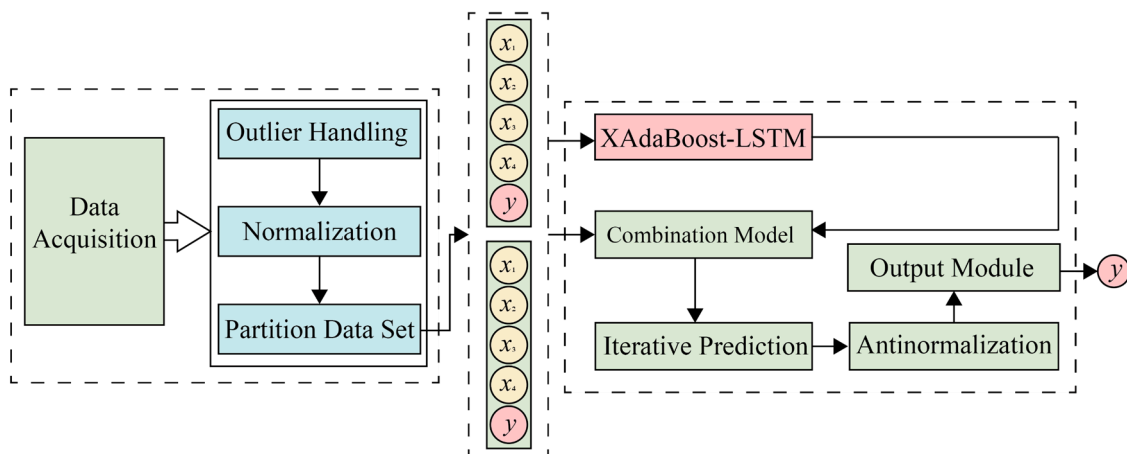


Figure 4. Model prediction flow chart.

4. Analysis of Results

4.1. Raw Data

To verify the feasibility of the XAdaBoost algorithm improvement strategy and the accuracy and stability of the XAdaBoost–LSTM algorithm modeling prediction, the turbot breeding data measured by the intensive seawater circulation control system was used as the modeling data in the experiment. The data collection period was from 08:00 to 18:00 every day from April to June 2017, with a time interval of 2 h; a total of 236 data groups were collected. In the experiment, the corresponding sensor module was used to collect

water quality parameters such as temperature, dissolved oxygen, pH value, and electrical conductivity of each group, and the ammonia nitrogen concentration of each group was measured using the chemical reagent method. Among them, the temperature sensor model used was a PT100-type platinum resistance contact sensor; the sensor range was $-5-60\text{ }^{\circ}\text{C}$. The dissolved oxygen concentration sensor used was the FDO 700 IQ sensor of the German WTW company; the measurement range of the dissolved oxygen sensor was $0-20\text{ mg/L}$. The pH sensor used was a Senso-Lyt 700 IQ SEA sensor from WTW, Germany, which had a measurement range of $2-14$. The conductivity sensor used was the TetraCon700 IQ sensor of WTW company in Germany; the measurement range was $0.1-500\text{ ms/cm}$. Of course, there are some accuracy errors in the actual data acquisition process of the sensor that could be avoided; for example, the temperature sensor had an error of $\pm 0.5\%$, and the dissolved oxygen concentration also had an error of $\pm 0.01\text{ mg/L}$. However, these errors are minor compared to the numerical value, and you can make negligible assumptions. Among them, the measured value of ammonia nitrogen concentration was the predicted quantity, which was the model output, and the other water quality factors were used as auxiliary variables as the model input. Some experimental data are shown in Table 1.

Table 1. Partial data of water quality parameters of turbot culture.

Moment	Temperature $^{\circ}\text{C}$	Dissolved Oxygen ($\text{mg}\cdot\text{L}^{-1}$)	pH	Electrical Conductivity ($\text{ms}\cdot\text{cm}^{-1}$)	Ammonia Nitrogen Concentration ($\text{mg}\cdot\text{L}^{-1}$)
8:00	14.9	8.2	7.85	41.1	0.18
10:00	14.7	8.26	7.85	41.1	0.21
12:00	14.5	8.27	7.91	41.2	0.23
...
18:00	14.3	8.33	7.91	41.1	0.24

Table 1 shows that the turbot culture data contained four input features, temperature, dissolved oxygen, pH, and conductivity, and one output feature, ammonia nitrogen concentration.

4.2. Data Pre-Processing

The 3σ rule and median filtering method have been used to detect outliers in previous research in the lab [22]. The 3σ criterion, where σ stands for standard deviation, measures how far a number in a data set deviates from its mean. In typically or nearly normally distributed data sets, approximately 68.27% of the data is within the mean $\pm 1\sigma$, 95.45% of the data is within the mean $\pm 2\sigma$, and 99.73% of the data is within the mean $\pm 3\sigma$. Therefore, if the value of a data point is outside the range of the mean $\pm 3\sigma$, we can consider this data point as an outlier.

Since the input variables are not unique and have different values and units, it is necessary to normalize the data to eliminate the influence of the magnitude [23]. There are two standard normalization methods: the first is linear function normalization, which converts all input values into values in the interval $[0, 1]$. The second method is zero-mean normalization, which transforms the data set into a data set with zero mean and one variance. This experiment uses the first normalization method, i.e., linear function normalization, with the following equation:

$$X_N = \frac{X - X_{MIN}}{X_{MAX} - X_{MIN}} \quad (16)$$

where X_N represents the normalized data, X_{MIN} represents the smallest number in a feature data set, and X_{MAX} represents the largest number in a feature data set. For the missing values among them, linear interpolation is used to fill the missing data. If there is a missing value at time t , a linear interpolation operation is performed using the known

data before and after time t , and the result is supplemented as the value at time t with the following formula:

$$x_t = x_{t_q} + \frac{(x_{t_h} - x_{t_q})(t - t_q)}{t_h - t_q} \quad (17)$$

where x_t represents the missing value at the time t , x_{t_q} represents the most recent known data before time t , and x_{t_h} represents the most recent known data after time t .

4.3. Operating Environment and Evaluation Criteria

The ammonia nitrogen concentration prediction is a multivariate prediction model that uses SciPy(1.5.4), NumPy(1.14.3), Matplotlib(2.2.2), Pandas(0.23.0), and Scikit-learn(0.24.2) libraries in machine learning and Theano, TensorFlow(2.0), and Keras(2.3.1) libraries in deep learning. This experiment used the LSTM model as a sub-model, which contained 150 neurons in the input layer, 50 neurons in the hidden layer, and 1 neuron in the output layer. In order to prevent the occurrence of overfitting, the value of Dropout was set to 0.3, the Optimizer selected the ADAM algorithm to optimize, and the loss function selected the mean square error (MAE) to calculate. After experimental verification in the XAdaBoost model, the number of iterations was finally selected as three, and the model used for iteration was the LSTM model, in which the loss function was linear. The experiment used the temperature, conductivity, pH value, and dissolved oxygen concentration as the feature data input, and the ammonia nitrogen concentration was used as the output to train the model. Before training the model, the data was divided into 76% of the training set and 24% of the test set. In the laboratory culture process, temperature, electrical conductivity, water pH and dissolved oxygen concentration changes all affect the change of ammonia nitrogen concentration in the water. Therefore, in aquaculture, temperature, electrical conductivity, water pH, dissolved oxygen concentration, and ammonia nitrogen concentration interact with each other, and the relationship is coupled. Therefore, in this experiment, temperature, electrical conductivity, water pH, and dissolved oxygen concentration closely related to ammonia nitrogen concentration were used as inputs.

In evaluating the model, the evaluation criteria used in this paper were as follows: the root mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE) [24].

The formulas of the three evaluation criteria are shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i^* - y_i)^2}{n}} \quad (18)$$

$$MAE = \frac{\sum_{i=1}^n |y_i^* - y_i|}{n} \quad (19)$$

$$MAPE = \frac{\sum_{i=1}^n \left| \frac{y_i^*}{y_i} \right|}{n} \quad (20)$$

where y_i^* represents the predicted value of the model, y_i represents the true value, and n represents the number of samples. $RMSE$, MAE , $MAPE$ represent three evaluation criteria containing information on the error between the predicted and true values as well as the sample proportion.

4.4. Experimental Analysis and Evaluation

Since the last iteration of the XAdaBoost algorithm was to be based on the comparison of multiple error rate results, the number of iterations $T = 3$ was finally selected in this paper after several experiments; the last one was used as the corrected iteration after improvement. The error rate range d in the experiment was selected based on the distribution of the first

2 error rates in the model. A comparison of the LSTM prediction model with the AdaBoost-LSTM prediction model and the prediction results are shown in Figure 5.

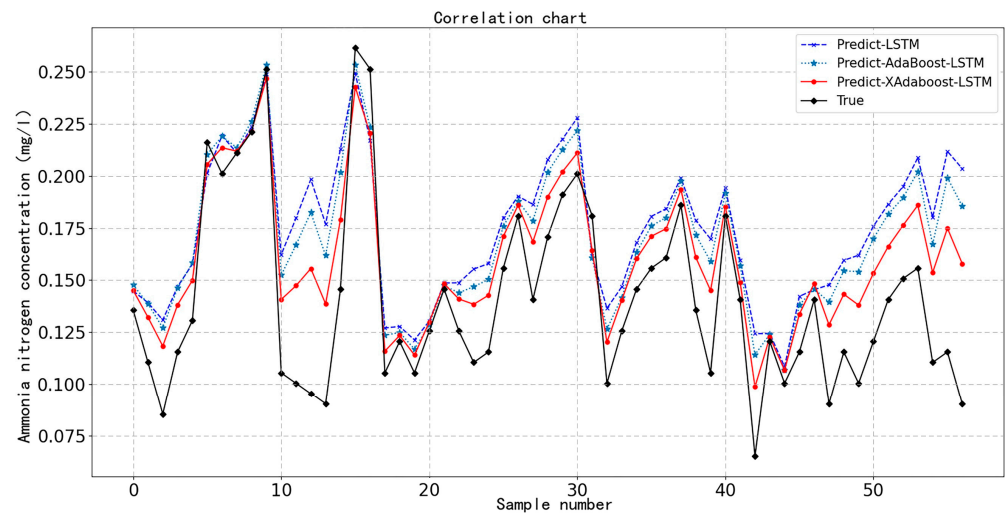


Figure 5. Comparison chart of experimental results.

In the figure, the XAdaBoost-LSTM is the prediction model of the improved AdaBoost algorithm for LSTM enhancement and compares the two models of LSTM and AdaBoost-LSTM. Firstly, it can be seen that the AdaBoost algorithm could improve the accuracy of the LSTM prediction model through iteration, which was better in some details compared to the LSTM; secondly, it can be seen that the XAdaBoost-LSTM prediction graph is more obvious in the fitting effect, which verifies the feasibility of the improved AdaBoost algorithm. It can be seen that the change trends before and after the improvement are very similar, but its results in some details are better and more stable compared to the improved one. In Figure 5, when the XAdaBoost-LSTM model predicts the low value of ammonia nitrogen concentration, the prediction accuracy is insufficient, but compared with the other two models, the accuracy is the highest when the low value of ammonia nitrogen concentration is predicted.

Meanwhile, a comparison experiment was set up to compare the effectiveness of the proposed method with MLP and another LSTM method using CNN improvement; the prediction results are shown in Figure 6.

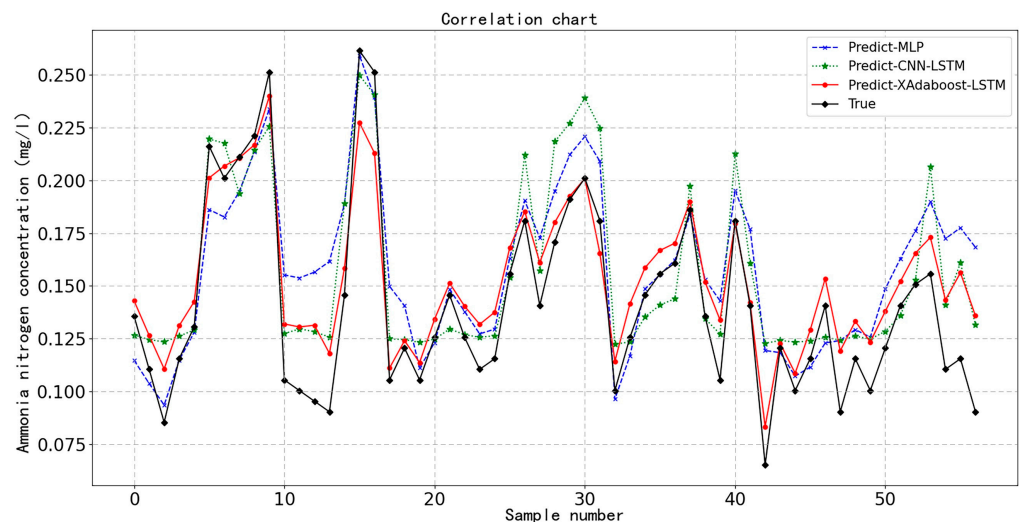


Figure 6. Comparison chart of multi-model experimental results.

As can be seen from the figure, adding the CNN network before the recurrent neural network to extract the features of the data, the CNN–LSTM model was better in some details, but the overall fitting effect was not as good as the XAdaBoost–LSTM model; the MLP model had abrupt variability in the prediction points in the figure several times, which was because the network did not consider the temporality of the sequence. On the contrary, the LSTM model with XAdaBoost boost had high error tolerance, so it had better results in dealing with the task of temporality and the presence of errors.

In this paper, three sub-models were generated after iteration, and each sub-model handled data with different weights. To facilitate the observation of features, some data points were intercepted to show the prediction effects of the three sub-models, as shown in Figures 7–9.

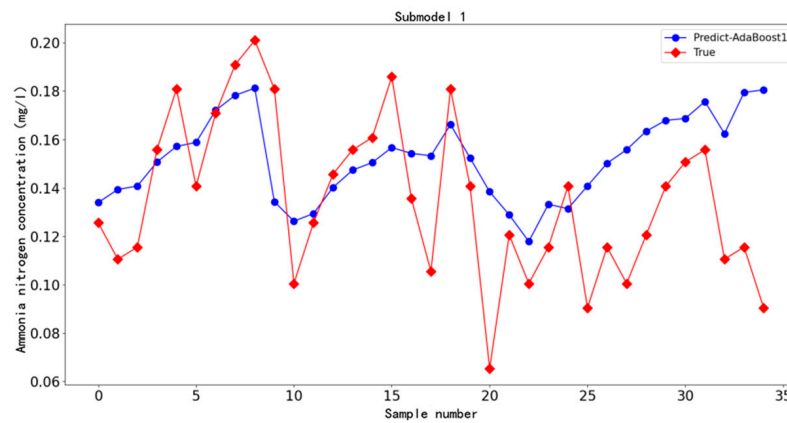


Figure 7. Sub-model 1 result plot.

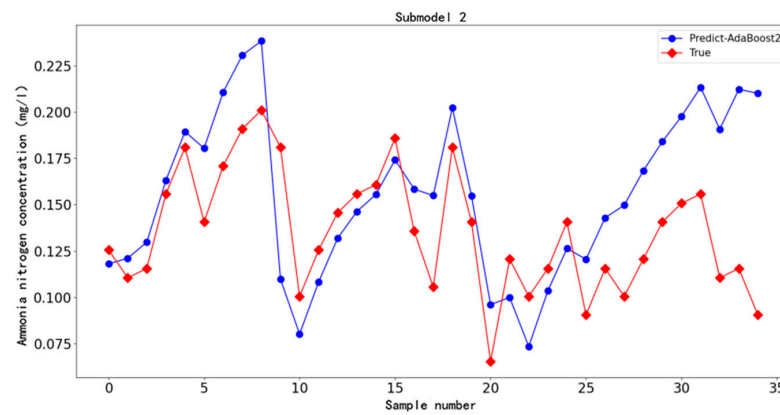


Figure 8. Sub-model 2 result plot.

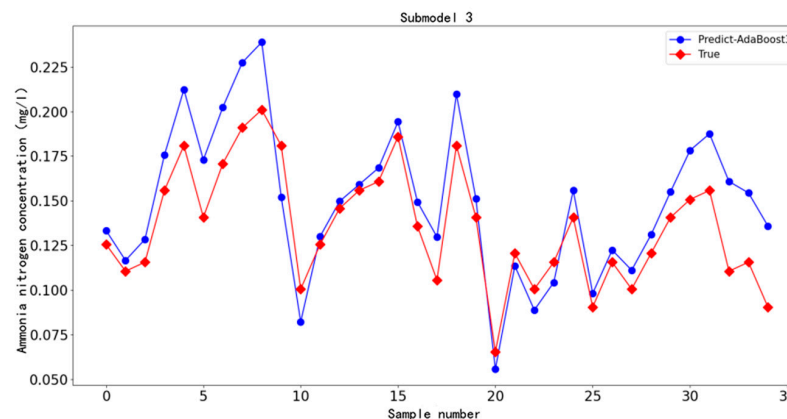


Figure 9. Sub-model 3 result plot.

Figure 7 shows the first generated sub-model 1; it can be seen that the trend of most points in the figure is not apparent and has a significant error. After the first weak model was obtained, the data weights were reassigned according to the prediction results.

Figure 8 shows sub-model 2 generated after iteration. The weights of the poorly predicted data were enlarged based on the results predicted by sub-model 1, so the model was more accurate in dealing with the points that did not work well in sub-model 1. At the same time, the image was better fitted because the weights of other points are reduced.

Figure 9 shows the sub-model 3 generated after the iteration, which was also the corrective iteration. The model was built based on the first two iterations, and by comparing the magnitude of the error rate, the weights of the cases where both error rates were not satisfied were set to 0. After zooming in again for the data with poor prediction results, it can be seen that the distribution of some points in the weak model 3 differs from sub-models 1 and 2, but the overall effect was better than the first two prediction results.

In the last iteration, the error rate d was selected as 0.058 by the results of the first two model processing results of the data. If the first two sub-model prediction results' error rate was greater than d , the weight was set to 0 in the third correction iteration process. The model error rate distribution is shown in Figure 10.

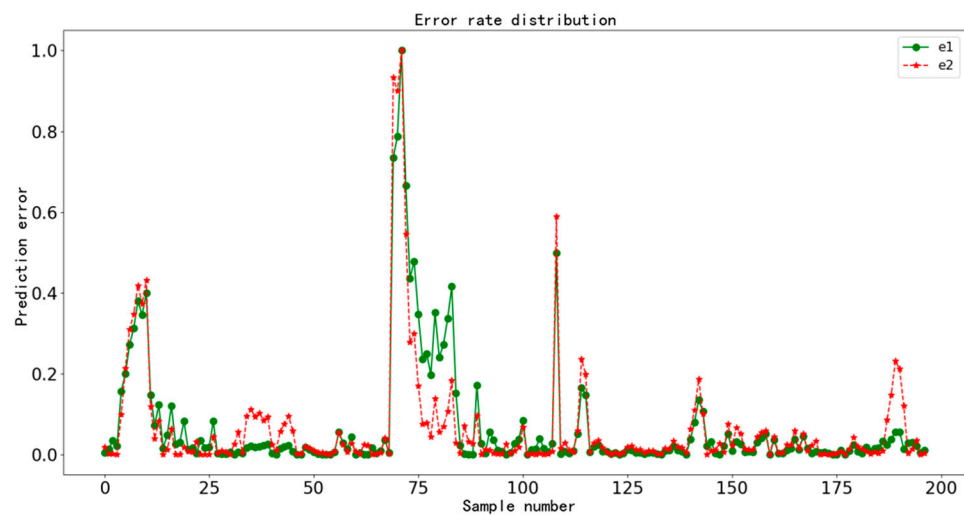


Figure 10. Two model prediction error rate distributions.

e_1 , e_2 were the first and second iteration error rate distributions; we can see that through the change of data weights most of the data in the two prediction results error rate difference was not large, but there were also noticeable differences in the second rather than the first good point, which provides a the basis that AdaBoost can improve the accuracy rate. There are also some points where the error rate became larger with the amplification of the data, and the combination of different data changes could effectively improve the stability. Some of the data in the figure have large error rates on both occasions, and treating such data as error data and disregarding their effects in the last iteration can effectively improve model stability and accuracy. The same three metrics, RMSE, MAE, and MAPE, were used in this experiment to assess the accuracy and stability of the model. The results are shown in Table 2.

Table 2. Model evaluation index.

Models	RMSE	MAE	MAPE
MLP	0.0634	0.0523	41.4199
LSTM	0.0601	0.0456	38.1255
CNN-LSTM	0.0489	0.0412	31.9996
ADABOOST-LSTM	0.0441	0.0348	31.7202
XADABOOST-LSTM	0.0352	0.0276	23.0314

Table 2 shows the average of the results of 20 experiments. After AdaBoost’s iterative enhancement to LSTM, the three indicators were significantly reduced so that AdaBoost could improve the accuracy of the model prediction. The three indicators of the XAdaBoost–LSTM model were lower than those of AdaBoost, which could prove the effect of the improved algorithm and the effect of the model. These three indicators could intuitively reflect that the indicators of the XAdaBoost–LSTM prediction model are significantly lower than those of other machine learning models and artificial intelligence models such as MLP.

In addition, the GRU series models have fewer parameters, which makes the model widely used in the field of aquaculture. Therefore, we also used the evaluation criteria to verify that the prediction accuracy of XAdaBoost–LSTM was higher than that of the GRU series models. Since its introduction, the CNN model has been widely used and achieved good performance in different fields. Therefore, we also added the CNN model to make the supplementary experiment more perfect. The comparison of evaluation criteria of the supplementary model is shown in Table 3.

Table 3. Supplementary model evaluation indicators.

Models	RMSE	MAE	MAPE
CNN	0.0685	0.0610	52.8130
GRU	0.0546	0.0424	37.307
CONV-GRU	0.0416	0.0313	26.4624
XADABOOST–LSTM	0.0352	0.0276	23.0314

It can be seen from Table 3 that the XAdaBoost–LSTM model had more accurate prediction performance than the GRU series models with fewer parameters and computational complexity and the CNN model with a wide range of applications.

4.5. Model Stability Analysis

To verify that the XAdaBoost algorithm and AdaBoost algorithm could improve the stability of the models, the stability and robustness of the models were reacted by observing the changes of RMSE, MAE, and MAPE for 20 predictions of the LSTM, AdaBoost–LSTM, and XAdaBoost–LSTM models, as shown in Figures 11–13.

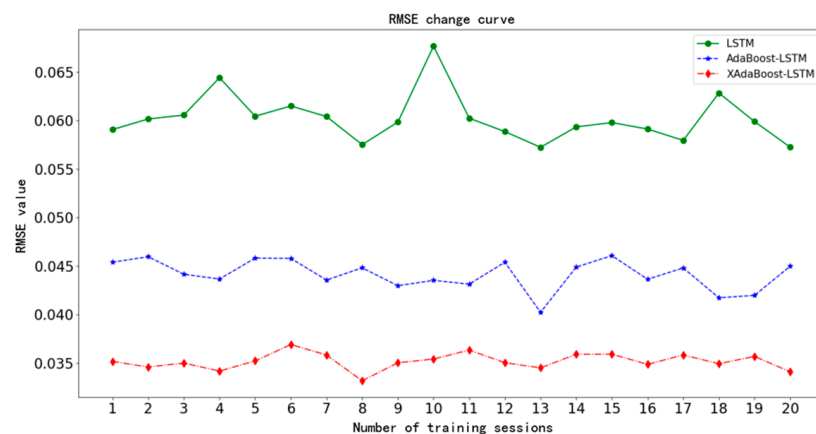


Figure 11. RMSE change chart.

It can be seen from the figure that the changes in the three indicators of the prediction model were smoother than those of the LSTM model after the iterative enhancement of XAdaBoost and AdaBoost during multiple experiments. The changes of XAdaBoost indicators were the smoothest and the changes were the smallest, indicating that the XAdaBoost–LSTM prediction model was more stable. It proves that the XAdaBoost algorithm can effectively solve the problems of poor single prediction robustness and low stability of traditional prediction models.

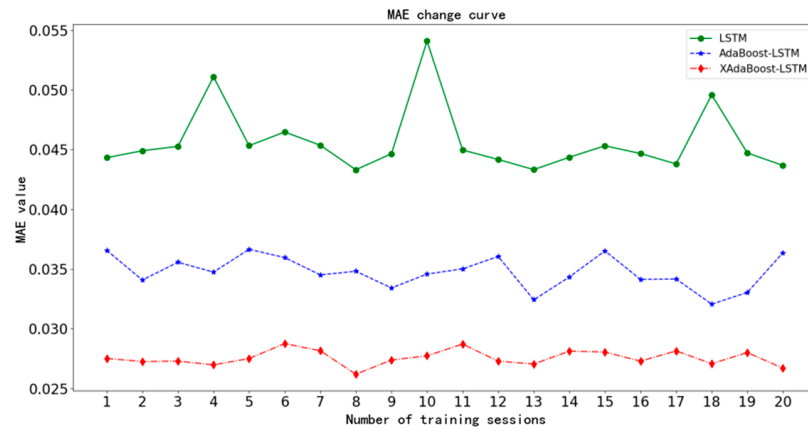


Figure 12. MAE change chart.

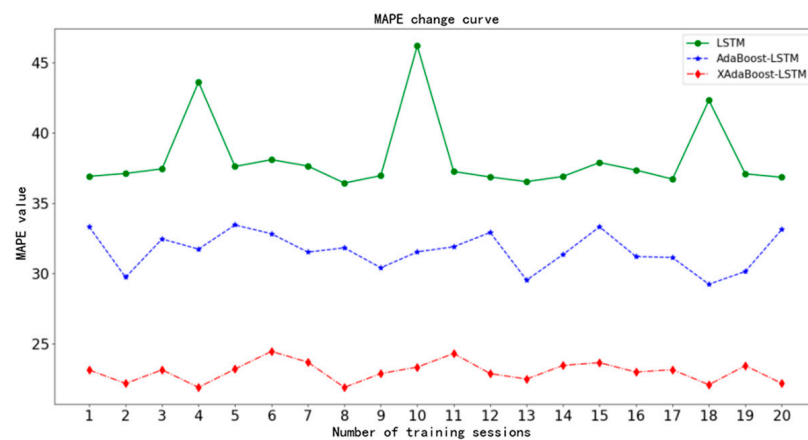


Figure 13. MAPE change chart.

AdaBoost improves model stability and accuracy from the perspective of data weights, but the different number of iterations will affect the model performance. The XAdaBoost–LSTM ammonia nitrogen concentration prediction model is the traditional Adaboost–LSTM algorithm when the number of iterations is less than 3, and corrective iterations can be introduced when $T > 3$. The RMSE, MAE, and MAPE for different iterations are shown in Table 4.

Table 4. The results of XAdaBoost–LSTM in different iterations.

Number of Iterations	RMSE	MAE	MAPE
1	0.0591	0.0443	38.1255
2	0.0441	0.0348	31.7202
3	0.0352	0.0276	23.0314
4	0.0426	0.0344	28.1742

The table shows the average results of 20 experiments. The model works best at the third iteration, and the model prediction effect starts to decrease when the number of iterations continues to increase, because the AdaBoost integrated learning model keeps amplifying the error data when assigning weights to the training data each time, so after the model reaches the optimum, increasing the number of iterations will only increase the error rate of the model, and the number of iterations will change depending on the data type. XAdaBoost introduces corrective iterations, but the error accumulation is not eliminated, so the effect of the model will still decrease with more iterations.

5. Conclusions

To solve the problem that a single LSTM ammonia nitrogen prediction model is susceptible to sudden changes and error information in the prediction process, which leads to low prediction accuracy and robustness, an XAdaBoost integrated algorithm with an improved combination approach was proposed to consider the feature that the combined learning algorithm can build multiple sub-models; the XAdaBoost–LSTM ammonia nitrogen concentration prediction model was proposed. The ammonia nitrogen concentration prediction sub-model was established using LSTM, and the data weights were updated by the XAdaBoost integrated learning algorithm so that the model could fully consider the sample characteristics. The distribution of the sub-model on different data was analyzed, and the model changes were compared for multiple experiments. It was verified that XAdaBoost could improve the accuracy and stability of the prediction model, and the effect was better than AdaBoost. The algorithm had universal applicability to different models, and the proposed method was verified to be more effective than MLP and CNN–LSTM in the task of ammonia nitrogen concentration prediction through comparison experiments, and more suitable for the design of an ammonia nitrogen concentration prediction system in aquaculture water quality.

Applying the prediction model in this paper to the aquaculture system is of great significance for formulating the regulation strategy in advance. By setting the prediction time step, the change curve of ammonia nitrogen concentration can be predicted in the future. By analyzing the change curve of ammonia nitrogen concentration, the safe range of ammonia nitrogen concentration corresponded to determine the exceedance time of ammonia nitrogen concentration, and the reasons for the exceedance of ammonia nitrogen concentration were analyzed. Suppose the ammonia nitrogen concentration in the water body increases due to the accumulation of food residues and excreta in the aquaculture pond without water change operation for a long time. In that case, management strategies such as water change and silt removal should be adopted at that point. If the water change interval is short, the concentration of ammonia nitrogen is also in the excessive range, which is the method of oxygen increase that can be used to reduce the concentration of ammonia nitrogen. High oxygen content means that ammonia nitrogen has little toxic effect on aquaculture. In addition, in this case, it is necessary to check whether the pH value of the water and the temperature are within the reasonable range for the survival of the aquaculture. When the pH value is higher, the ammonia nitrogen concentration increases more rapidly; otherwise, it is slower. When the temperature is too high, it will also increase the concentration of ammonia nitrogen in the water. When the above methods have not solved the problem of unreasonable concentration of ammonia nitrogen, the method of drug delivery can be used to reduce the concentration of ammonia nitrogen. If the previous methods have failed to address the issue of excessive concentration of ammonia nitrogen, drug delivery can be used to reduce its concentration. However, applying this model in aquaculture systems poses certain challenges. The model's prediction accuracy can be impacted by large-scale data, and it is yet to be determined whether the model can maintain stability and prediction accuracy across different aquaculture species. The application of the prediction model in aquaculture management systems involves designing the function and relationship of each module and encapsulating all modules into a platform system, which includes data acquisition and processing, prediction, output, and storage modules. Water quality data is collected by sensors and transmitted to the database, and a prediction model module is established in the system. The data is then inputted into the prediction module to obtain the predicted value.

In the future, we will further use the data sets collected by different aquaculture systems to verify the model, proving that the model also has good prediction accuracy and stability in different systems. In addition, we will use feature construction, merging data, and other methods to increase the number of features of aquaculture water quality in the data and combine the model with the actual Internet of Things devices to further improve the model's generalization. Finally, we will apply the model to the water quality management system to assess the practice.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W., X.L. and W.W.; software, Y.W., D.X., X.L. and W.W.; data curation, D.X.; writing—original draft, D.X.; writing—review & editing, X.L.; visualization, W.W.; supervision, Y.W. and W.W.; project administration, W.W.; funding acquisition, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Open Project of Key Laboratory of Environment Controlled Aquaculture, Ministry of Education, China (Project Number: 202314), the General Scientific Research Projects of Liaoning Provincial Department of Education, China (Project Number: JYTMS20230489), and the data came from the intensive mariculture system of the modeling laboratory of Dalian Ocean University.

Data Availability Statement: The data will be made available by the authors on request.

Acknowledgments: Special thanks to the editors and the four reviewers for their constructive comments.

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

References

1. Wang, J.X.; Li, J.Y.; Li, S.J.; Shi, Y.Q.; Gao, Y.; Ji, G.J. Exploration of aquaculture standardization in China. *China Aquac.* **2019**, *4*, 31–34.
2. Zhang, C.L.; Xu, H.; Wang, S.M.; Liu, H.; Chen, X.J. Current situation and considerations on the development of deep-sea fisheries based on large-scale fisheries platforms. *China Agron. Bull.* **2020**, *36*, 152–157.
3. Nagaraju, T.V.; Sunil, B.M.; Chaudhary, B.; Prasad, C.D.; Gobinath, R. Prediction of ammonia contaminants in the aquaculture ponds using soft computing coupled with wavelet analysis. *Environ. Pollut.* **2023**, *331 Pt 2*, 121924. [[CrossRef](#)] [[PubMed](#)]
4. Le, P.T.T.; Boyd, C.E. Comparison of Phenate and Salicylate Methods for Determination of Total Ammonia Nitrogen in Freshwater and Saline Water. *J. World Aquac. Soc.* **2012**, *43*, 885–889. [[CrossRef](#)]
5. Xu, G.; Xu, G.; Wei, H.; Wei, H.; Wang, J.; Wang, J.; Chen, X.; Chen, X.; Zhu, B.; Zhu, B. A Local Weighted Linear Regression (LWLR) Ensemble of Surrogate Models Based on Stacking Strategy: Application to Hydrodynamic Response Prediction for Submerged Floating Tunnel (SFT). *Appl. Ocean. Res.* **2022**, *125*, 103228. [[CrossRef](#)]
6. Ottaviani, F.M.; De Marco, A. Multiple Linear Regression Model for Improved Project Cost Forecasting. *Procedia Comput. Sci.* **2022**, *196*, 808–815. [[CrossRef](#)]
7. Chen, W.B.; Liu, W.C. Artificial neural network modeling of dissolved oxygen in the reservoir. *Environ. Monit. Assess.* **2014**, *186*, 1203–1217. [[CrossRef](#)] [[PubMed](#)]
8. Nong, X.; Lai, C.; Chen, L.; Shao, D.; Zhang, C.; Liang, J. Prediction modeling framework comparative analysis of dissolved oxygen concentration variations using support vector regression coupled with multiple feature engineering and optimization methods: A case study in China. *Ecol. Indic.* **2023**, *146*, 109845. [[CrossRef](#)]
9. Zhao, S.; Gui, F.L.; Liu, H.Q. Prediction of nitrogen concentration in Taihu Lake based on AdaBoost machine learning model. *China Rural. Water Hydropower* **2022**, *6*, 24–28.
10. Belavadi, S.V.; Rajagopal, S.; Ranjani, R.; Mohan, R. Air Quality Forecasting using LSTM RNN and Wireless Sensor Networks. *Procedia Comput. Sci.* **2020**, *170*, 241–248. [[CrossRef](#)]
11. Farhi, N.; Kohen, E.; Mamane, H.; Shavitt, Y. Prediction of wastewater treatment quality using LSTM neural network. *Environ. Technol. Innov.* **2021**, *23*, 101632. [[CrossRef](#)]
12. Huan, J.; Li, H.; Li, M.; Chen, B. Prediction of dissolved oxygen in aquaculture based on gradient boosting decision tree and long short-term memory network: A study of Chang Zhou fishery demonstration base. *Comput. Electron. Agric.* **2020**, *175*, 105530. [[CrossRef](#)]
13. Yan, K.; Li, C.; Zhao, R.; Zhang, Y.; Duan, H.; Wang, W. Predicting the ammonia nitrogen of wastewater treatment plant influent via integrated model based on rolling decomposition method and deep learning algorithm. *Sustain. Cities Soc.* **2023**, *94*, 104541. [[CrossRef](#)]
14. Ruma, J.F.; Adnan, M.S.G.; Dewan, A.; Rahman, R.M. Particle swarm optimization based LSTM networks for water level forecasting: A case study on Bangladesh river network. *Results Eng.* **2023**, *17*, 100951. [[CrossRef](#)]
15. Busari, G.A.; Lim, D.H. Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Comput. Chem. Eng.* **2021**, *155*, 107513. [[CrossRef](#)]
16. Xiao, C.; Chen, N.; Hu, C.; Wang, K.; Gong, J.; Chen, Z. Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach. *Remote Sens. Environ.* **2019**, *233*, 111358. [[CrossRef](#)]
17. Wadud, M.A.H.; Kabir, M.M.; Mridha, M.F.; Ali, M.A.; Hamid, M.A.; Monowar, M.M. How can we manage Offensive Text in Social Media—A Text Classification Approach using LSTM-BOOST. *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100095. [[CrossRef](#)]
18. Shi, W.; Hu, L.; Lin, Z.; Zhang, L.; Wu, J.; Chai, W. Short-term motion prediction of floating offshore wind turbine based on multi-input LSTM neural network. *Ocean Eng.* **2023**, *280*, 114558. [[CrossRef](#)]
19. Qiu, K.; Li, J.; Chen, D. Optimized long short-term memory (LSTM) network for performance prediction in unconventional reservoirs. *Energy Rep.* **2022**, *8*, 15436–15445. [[CrossRef](#)]

20. Zhou, L.; Boyd, C.E. Comparison of Nessler, phenate, salicylate, and ion selective electrode procedures for determination of total ammonia nitrogen in aquaculture. *Aquaculture* **2016**, *450*, 187–193. [[CrossRef](#)]
21. Li, Y.; Li, R. Predicting ammonia nitrogen in surface water by a new attention-based deep learning hybrid model. *Environ. Res.* **2023**, *216 Pt 3*, 114723. [[CrossRef](#)] [[PubMed](#)]
22. Wang, W.; Guo, G. Soft-sensing model of ammonia nitrogen concentration in perception based on random configuration network. *Trans. Chin. Soc. Agric. Mach.* **2020**, *51*, 214–220.
23. Nasiri, A.; Yoder, J.; Zhao, Y.; Hawkins, S.; Prado, M.; Gan, H. Pose estimation-based lameness recognition in broiler using CNN-LSTM network. *Comput. Electron. Agric.* **2022**, *197*, 106931. [[CrossRef](#)]
24. Yu, H.; Yang, L.; Li, D.; Chen, Y. A hybrid intelligent soft computing method for ammonia nitrogen prediction in aquaculture. *Inf. Process. Agric.* **2021**, *8*, 64–74. [[CrossRef](#)]

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