Journal of Environmental Informatics 41(2) 104-117 (2023)

Journal of Environmental Informatics

www.iseis.org/jei

Time-Series Forecasting of Chlorophyll-a in Coastal Areas Using LSTM, GRU and Attention-Based RNN Models

S. S. Wu[1], Z. H. Du[1, 2 *], F. Zhang[1, 2], Y. Zhou[3, 4], and R. Y. Liu[1, 2]

[1] *School of Earth Sciences, Zhejiang University, Hangzhou, Zhejiang 310027, China*
[2] *Zhejiang Provincial Key Laboratory of Geographic Information Science, Hangzhou, Zhejiang 310028, China*
[3] *Zhejiang Fisheries Technical Extension Center, Hangzhou, Zhejiang 310013, China*
[4] *Zhejiang Province Academy of Marine Science, Hangzhou, Zhejiang 310012, China*

Received 06 March 2018; revised 03 August 2019; accepted 11 January 2020; published 15 May 2023

ABSTRACT. The chlorophyll-a (Chl-a) concentration is commonly considered as the main indicator of phytoplankton biomass in coastal waters. Forecasting and understanding the status of Chl-a is beneficial to coastal ecosystem management and is an important emergency management measure for algae blooms. To obtain accurate predictions, the long short-term memory neural network (LSTM) and gated recurrent unit neural network (GRU) were implemented for Chl-a forecasting, and based on the LSTM and GRU units, two simplified attention-based encoder-decoder recurrent neural network (AEDRNN) models were also developed for time series predictions. The performance of the proposed models was compared with that of the auto-regressive integrated moving average (ARIMA), multilayer perceptron (MLP) and Elman recurrent neural network (ERNN) models by experimentally generating multi-step-ahead predictions using a dataset in the Zhejiang coastal areas of China. The results demonstrated that the LSTM, GRU and AEDRNN models significantly outperformed the ARIMA, MLP and ERNN models according to multiple statistical indicators. Moreover, the AEDRNN models were superior to the LSTM and GRU models, especially for middle-term predictions. In addition, the AEDRNN model with LSTM units was more robust than the AEDRNN model with GRU units in terms of accuracy and stability; therefore, it was considered to be the best model for Chl-a forecasting.

Keywords: attention-based encoder-decoder recurrent neural network, algal blooms, chlorophyll-a, gated recurrent unit neural network, long short-term memory neural network, time-series forecasting

1. Introduction

Algal blooms and eutrophication in estuaries and coastal areas have caused considerable damage to the usage of coastal resources and become serious water quality issues worldwide (Chen et al., 2015; Rajaee and Boroumand, 2015). The chlorophyll-a (Chl-a) concentration is commonly regarded as the main measurement of phytoplankton biomass, and is used to evaluate the trophic status and infer algal blooms of coastal waters (Samli et al., 2014; Wang et al., 2015).

Forecasting and understanding the status of phytoplankton biomass (Chl-a) is considered to be highly useful for coastal ecosystem management and algae bloom early warning. In recent decades, many studies have been conducted to develop and improve time series forecast models. Numerous models have been applied to the prediction of Chl-a and other water quality parameters to facilitate the development of management strategies for estuarine and coastal waters.

One of the most fundamental prediction models is the auto-regressive integrated moving average (ARIMA) model (Box and Jenkins, 1976). ARIMA and its variants (e.g., SARIMA and ARIMAX) have been widely applied to water quality forecasting in estuaries and coastal areas (Ahmad et al., 2001; Nicholls, 2012; Photphanloet et al., 2016). However, these models have unstable accuracy due to their linear representation of the nonlinear system (Zhang, 2003), such as the behaviours of algae blooms in nutrient-enriched water bodies, which involve the nonlinear physical, chemical, and biological processes (Wang et al., 2015).

Recently, feed-forward neural networks (FNNs) have achieved great success in algal blooms (Chl-a) prediction (Lee et al., 2003; Melesse et al., 2008; Samli et al., 2014; Wang et al., 2016). Although FNNs exhibit better performance than traditional linear models for the complexity of ecological phenomena, they present difficulties in modelling temporally dynamic systems because of their hierarchical network structures. To resolve this problem, recurrent neural networks (RNNs) that integrate a stateful memory mechanism into FNNs were introduced, and can be trained to learn time-varying patterns and thus are generally more powerful than FNNs for modelling complicated dynamic systems (Brunelli et al., 2006; Harada et al., 2013; Wang et al., 2015; Le et al., 2017).

* Corresponding author. Tel.: +0571-8827-3287.
E-mail address: duzhenhong@zju.edu.cn (Z. H. Du).

ISSN: 1726-2135 print/1684-8799 online

However, traditional RNNs suffering from the vanishing or exploding gradient problem are unable to learn long time dependencies (Hochreiter and Schmidhuber, 1997; Gers, 2001). To address this shortcoming, the long short-term memory (LSTM) RNN with input and output gates was proposed by Hochreiter and Schmidhuber. (1997) and improved with the forget gate by Gers. (2001). The LSTM model is well-known for its excellent ability to memorize long-term dependencies and has achieved great success for time-series predictions in various fields (Maknickas, 2012; Ma et al., 2015; Bao et al., 2017; Fu et al., 2017). Moreover, a simplified variant of the LSTM architecture called the gated recurrent unit (GRU) neural network was also proposed to accelerate training (Cho et al., 2014). Many studies have shown that the performance of the GRU model is better than traditional RNNs, and is comparable to LSTM (Cho et al., 2014; Ravuri and Stolcke, 2016; Fan et al., 2017).

The features that memorize long-term dependencies of the LSTM and GRU models are also desirable for conducting time-series prediction in the coastal environmental domain. However, the LSTM and GRU models are rarely applied to modelling the environmental dynamic processes in coastal areas, and their availability for Chl-a prediction needs to be further studied.

In addition, a new class of recurrent architecture based upon LSTM or GRU units, namely, encoder-decoder recurrent neural network (EDRNN) (Cho et al., 2014; Sutskever et al., 2014), has attracted a great deal of attention and has become popular due to its recent success in machine translation. Since the performance of the basic EDRNN declines rapidly as the input sentence becomes longer, an attention-based encoder-decoder recurrent neural network (AEDRNN) that employs an attention mechanism was proposed by Bahdanau et al. (2014), and their results showed that the AEDRNN significantly outperformed the conventional EDRNN in machine translation. Consequently, the state-of-the-art AEDRNN method is naturally considered for time series predictions, and a few studies have shown that the performance of the AEDRNN was superior to that of other RNNs and FNNs (Zaytar and El Amrani, 2016; Cinar et al., 2017; Qin et al., 2017).

To our knowledge, the applicability of the AEDRNN model in forecasting coastal environmental processes has not been investigated. Therefore, the purpose of this paper is to examine the capabilities of the LSTM, GRU and AEDRNN models for Chl-a predictions in coastal areas, and the main contributions include the (1) introduction of robust RNN models to address the long-term temporal dependencies of Chl-a prediction in coastal areas; (2) development of two simplified AEDRNN models based on the LSTM and GRU units for time-series forecasting of Chl-a; (3) comparative investigation to estimate the applicability of the LSTM, GRU and AEDRNN models and provision of a general guideline for choosing suitable RNN models for Chl-a predictions.

The structure of this paper is as follows. The descriptions and architectures of the prediction models are presented in Section 2. The study area and data analysis are introduced in Section 3. The implementation of the models applied for Chl-a predictions are shown in Section 4. The experimental results and discussions of these models are provided to evaluate the performance in Section 5. The conclusions are given in the last section.

2. Methods

In this study, univariate time series predictions of Chl-a are the main concern. Based on the discussion in Section 1, the time series prediction models mainly fall into two categories: linear and nonlinear models. Regarding the nonlinear models, we focus on the artificial neural network (ANN) models, which can be simply divided into FNNs and RNNs. To evaluate the performance of the LSTM, GRU and AEDRNN models, the ARIMA and multilayer perceptron (MLP) are selected as representatives of linear models and FNNs, respectively, and the Elman RNN (ERNN) is chosen as the baseline RNN model.

2.1. ARIMA Model

In an ARIMA model, a variable's predicted value is considered a linear combination of historical values plus error terms, and it mainly includes three parts: auto-regression (AR), integration (I), and moving average (MA). Therefore, this model can be presented as the ARIMA (p, d, q), where p, d, and q are the numbers of the AR terms, difference terms, and MA terms, respectively. For example, ARIMA (p, 0, q) represents a time series depending on p past observations and q past random terms. The form of this model is as follows:

$$y_t = c + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - ... - \theta_q \varepsilon_{t-q} \quad (1)$$

where y_t and ε_t represent the true value and random error at time t. The random error is considered to be identically and independently distributed. In addition, β_i, θ_i and c are parameters that need to be estimated.

2.2. Multilayer Perceptron

The MLP is employed as the representative FNN in this paper. The three-layer feed-forward MLP that includes one hidden layer along with an input layer and an output layer is the most commonly used structure (Faruk, 2010; Samli et al., 2014). A schematic diagram of the MLP used in this study is given in Figure 1(a), where X_i (i = 1, 2, …, t) are the input variables, such as the temporal sequence of Chl-a in our case; H_i (i = 1, 2, …, s) denotes the hidden layer's outputs; and Y_{t+1} represents the predicted value. Here, MLP is used to predict the next moment of the preceding temporal sequence so that only one neuron is needed in the output layer.

2.3. Elman RNN

The ERNN proposed by Elman (1990) is used as the baseline RNN model in this study. The fundamental structure of the ERNN is illustrated in Figure 1(b). In this network, the hidden layer's outputs are permitted to be fed back through a context layer. Each neuron of the hidden layer connects to one neuron in the context layer with a constant weight. Accordingly, the context layer contains a copy of the hidden layer's states one instant before. Commonly, hidden neurons utilize nonlinear activation functions, such as sigmoid and tanh formulas, whereas input, output and context neurons employ linear transfer functions.

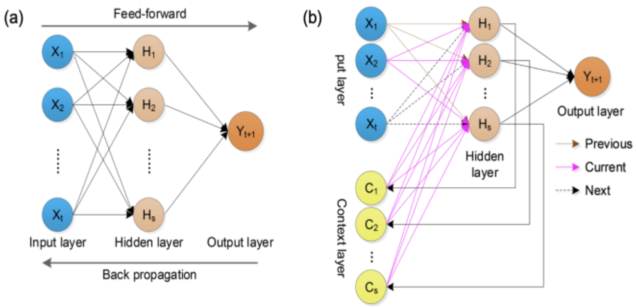

Figure 1. Network structures of the (a) MLP and (b) ERNN models.

2.4. LSTM and GRU RNNs

2.4.1. Long Short-Term Memory RNN

The LSTM that usually consists of one recurrent hidden layer along with an input layer and an output layer, was first presented by Hochreiter and Schmidhuber (1997). Unlike in conventional RNNs, its hidden layer's basic unit is a memory block for memorizing the temporal state. In the block, the input gate and output gate are used to determine the input and output, and the forget gate is to enable it to reset. In addition, peephole connections from the memory cell to the gates are created to filter the unwanted inputs or errors. The architecture is visualized in Figure 2(a).

The model input is displayed as $\boldsymbol{X} = (X_1, X_2, \ldots, X_t)$, and the output series is presented as $\boldsymbol{y} = (y_2, y_3, \ldots, y_{t+1})$, where t is the forecast period. In the case of Chl-a predictions, $\boldsymbol{X}$ is considered the historical data and $\boldsymbol{y}$ is the estimated Chl-a. To forecast the next moment's Chl-a, the prediction is iteratively calculated by equations (2) ~ (8):

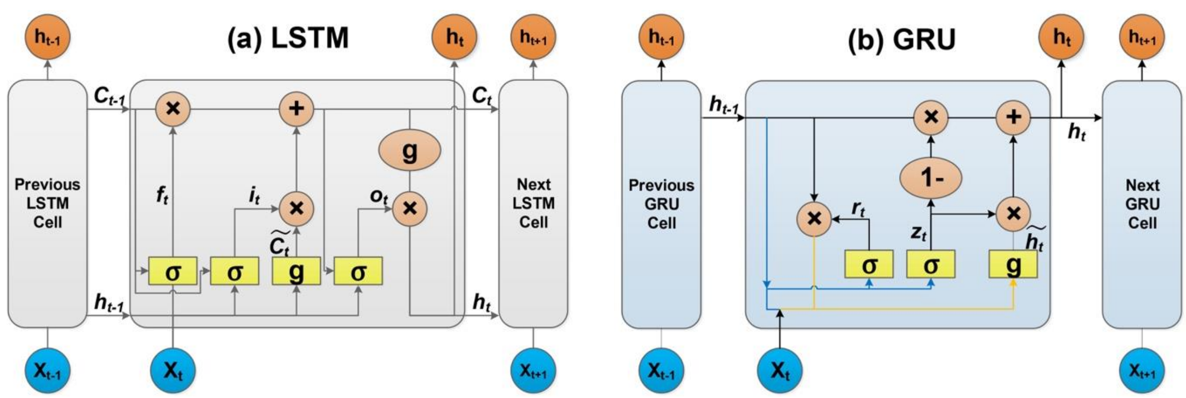

Figure 2. Network structures of the (a) LSTM and (b) GRU.

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \times [X_t, \boldsymbol{h}_{t-1}, \boldsymbol{C}_{t-1}] + \boldsymbol{b}_f) \tag{2}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \times g[X_t, \boldsymbol{h}_{t-1}, \boldsymbol{C}_{t-1}] + \boldsymbol{b}_i) \tag{3}$$

$$\tilde{\boldsymbol{C}}_t = g(\boldsymbol{W}_c \times [X_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_c) \tag{4}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t \times \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \times \tilde{\boldsymbol{C}}_t \tag{5}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \times [X_t, \boldsymbol{h}_{t-1}, \boldsymbol{C}_t] + \boldsymbol{b}_o) \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \times g(\boldsymbol{C}_t) \tag{7}$$

$$y_t = g(\boldsymbol{W}_y \times \boldsymbol{h}_t + \boldsymbol{b}_y) \tag{8}$$

where $\boldsymbol{f}_t$, $\boldsymbol{i}_t$ and $\boldsymbol{o}_t$ are the outputs of different gates, $\widetilde{\boldsymbol{C}_t}$ is the new cell state of the memory unit, $\boldsymbol{C}_t$ is the final cell state of the memory unit, $\boldsymbol{h}_t$ is the output hidden state of the memory unit; $\boldsymbol{W}_*$ and $\boldsymbol{b}_*$ are weight matrices and bias vectors, respectively; σ(·) is the sigmoid function, and its range is [0, 1]; and g(·) is the tanh function, and its range is [–1, 1]. The truncated back propagation through time (BPTT) and real time recurrent learning (RTRL) algorithms are applied to train the LSTM model through the gradient descent optimization method (Gers, 2001).

2.4.2. Gated Recurrent Unit RNN

The GRU neural network was presented by Cho et al. (2014), and it is similar to the LSTM but simpler to calculate and implement. The typical structure of GRU cells is shown in Figure 2(b). A typical GRU cell consists of two gates: the reset gate $\boldsymbol{r}_t$ and update gate $\boldsymbol{z}_t$. Similar to the LSTM cell, the hidden state output $\boldsymbol{h}_t$ and final prediction y_t at time t are computed using the hidden state of time t–1 and the input time series value at time t, which are presented in equations (9) ~ (13):

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z \times [X_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_z) \tag{9}$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \times [X_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_r) \tag{10}$$

$$\tilde{\boldsymbol{h}}_t = g(\boldsymbol{W}_h \times [X_t, \boldsymbol{h}_{t-1} \times \boldsymbol{r}_t] + \boldsymbol{b}_h) \tag{11}$$

$$\boldsymbol{h}_t = \tilde{\boldsymbol{h}}_t \times \boldsymbol{z}_t + \boldsymbol{h}_{t-1} \times (1 - \boldsymbol{z}_t) \tag{12}$$

$$y_t = g(\boldsymbol{W}_y \times \boldsymbol{h}_t + \boldsymbol{b}_y) \tag{13}$$

where the meanings of $\boldsymbol{W}_*$, $\boldsymbol{b}_*$, σ(·), and g(·) are the same as the LSTM formulas.

2.5. Attention-Based Encoder-Decoder RNN

The AEDRNN consists of three components: one is used to encode the input called encoder, one is to generate the output known as decoder, and the last one is used to denote the attention mechanism, which provides information from the inputs to generate output elements (Figure 3).

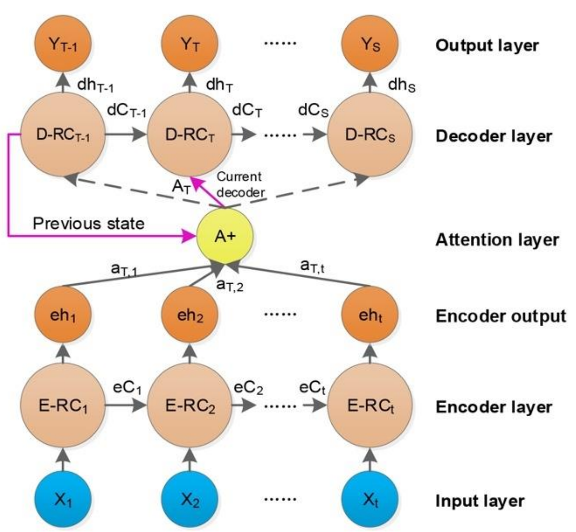

Figure 3. Network structures of the AEDRNN.

The encoder is essentially an RNN model that encodes the input sentences into feature vectors in machine translation. For time series predictions, given an input sequence $\boldsymbol{X} = (X_1, X_2, \ldots, X_t)$, the encoder is employed to produce the mapping from X_t to $[\boldsymbol{eC}_t, \boldsymbol{eh}_t]$ as follows:

$$[\boldsymbol{eC}_t, \boldsymbol{eh}_t] = f_{encoder\_cell}([\boldsymbol{eC}_{t-1}, \boldsymbol{eh}_{t-1}], X_t) \tag{14}$$

where $[\boldsymbol{eC}_t, \boldsymbol{eh}_t]$ is the state of the encoder at time step t and $f_{encoder\_cell}$ is a nonlinear transformation that depends on the considered RNN. In this paper, we examine the LSTM and GRU by using $f_{encoder\_cell}$. In machine translation modelling, Bahdanau et al. (2014) refined $f_{encoder\_cell}$ into bidirectional RNNs because the annotation of each word is related to both the preceding words and the following words. Since a time series prediction is generally regarded as a unidirectional process, a forward-directional RNN is adopted to capture the long-term dependencies.

To forecast the sequence output $\{y_T, y_{T+1}, \ldots, y_S\}$, another recurrent neural network is used to decode the encoded information. However, the encoder-decoder network's performance deteriorates quickly when the input sequence becomes longer. Therefore, a temporal attention mechanism is adopted after the encoder to adaptively choose relevant encoded hidden states. To be specific, the attention weight $\boldsymbol{\alpha}_{T,j}$ of each encoder hidden state $\boldsymbol{eh}_j$ at time T is calculated by the previous decoder states $[\boldsymbol{dC}_{T-1}, \boldsymbol{dh}_{T-1}]$ and $\boldsymbol{eh}_j$ as follows:

$$\boldsymbol{\beta}_{T,j} = f_{attention}(\boldsymbol{W}_\beta \times [\boldsymbol{dC}_{T-1}, \boldsymbol{dh}_{T-1}] + \boldsymbol{U}_\beta \times \boldsymbol{eh}_j + \boldsymbol{b}_\beta) \tag{15}$$

$$\boldsymbol{\alpha}_{T,j} = \exp(\boldsymbol{\beta}_{T,j}) / \sum_{k=1}^{t} \exp(\boldsymbol{\beta}_{T,k}) \tag{16}$$

where $\boldsymbol{W}_\beta$, $\boldsymbol{U}_\beta$ and $\boldsymbol{b}_\beta$ are the parameters to be learned; $f_{attention}$ is a nonlinear attention function, and the tanh function is applied. Then, the context vector $\boldsymbol{A}_T$ is calculated as a weighted sum of all the encoder hidden states $\{\boldsymbol{eh}_1, \boldsymbol{eh}_2, \ldots, \boldsymbol{eh}_t\}$:

$$\boldsymbol{A}_T = \sum_{j=1}^{t} \boldsymbol{\alpha}_{T,j} \times \boldsymbol{eh}_j \tag{17}$$

The context vector $\boldsymbol{A}_T$ of each time step is distinct. After obtaining this vector, we simplify the model of Bahdanau et al. (2014) by using $[\boldsymbol{dC}_{T-1}, \boldsymbol{dh}_{T-1}]$ and $\boldsymbol{A}_T$ to compute the decoder states:

$$[\boldsymbol{dC}_T, \boldsymbol{dh}_T] = f_{decoder\_cell}([\boldsymbol{dC}_{T-1}, \boldsymbol{dh}_{T-1}], \boldsymbol{A}_T) \tag{18}$$

According to our experience, this simplification is beneficial for enhancing computational efficiency without degrading performance. The form of the nonlinear transformation $f_{decoder\_cell}$ is the same as $f_{encoder\_cell}$.

Finally, a simplified AEDRNN architecture is developed, and the current output y_T is estimated as follows:

$$y_T = g(\boldsymbol{W}_y \times \boldsymbol{dh}_T + b_y) \tag{19}$$

where $\boldsymbol{W}_y$ and b_y are the weighted matrix and bias term, respecttively, and g(·) denotes the tanh function. In addition, because the GRU combines the states $[\boldsymbol{dC}_T, \boldsymbol{dh}_T]$ into one hidden state $\boldsymbol{dh}_T$, the states $[\boldsymbol{dC}_T, \boldsymbol{dh}_T]$ in equations (14) ~ (18) should be replaced by $\boldsymbol{dh}_T$ when using GRU cells.

2.6. Model Evaluation

The performance of the models is assessed using the following metrics: coefficient of determination (R^2), which measures the variability of the predicted data; and the mean absolute percentage error (MAPE) and root mean square error (RMSE), which estimate residual errors. The indicators are presented as equations (20) ~ (22):

$$R^2 = 1 - \sum_{i=1}^{N} (y_{oi} - y_{pi})^2 / \sum_{i=1}^{N} (y_{oi} - \bar{y}_o)^2 \tag{20}$$

$$MAPE = 1 / N \times \sum_{i=1}^{N} \left| y_{oi} - y_{pi} \right| / y_{oi} \times 100\% \tag{21}$$

$$RMSE = \sqrt{\sum_{i=1}^{N} (y_{oi} - y_{pi})^2 / N} \tag{22}$$

where y_o and y_p are the observed and predicted values, $\bar{y}_o$ is the mean of the observations, and N is the total count.

3. Study Area and Dataset

3.1. Study Area

The northern coastal area of Zhejiang Province situated in the East China Sea is chosen as the research area (Figure 4). This region experiences frequent human impacts and is the largest fishery of China. In recent decades, owing to the booming offshore shipping industry, this area has obtained considerable economic advantages. In addition, this area is located near the estuaries of the Qiangtang River and Yangtze River, which increases the complexity of the ecosystem. The high population density and economic activity have led to serious eutrophication in the region (Zhang et al., 2017a). Because of the highly complex and nonlinear characteristics of the marine ecological environment, addressing the eutrophication and algal blooms in the study area is urgent and challenging.

3.2. Dataset

Supported by the Zhejiang ocean projects, the Zhejiang Province Ocean and Fishery Bureau (ZJOFB) launched two water quality buoys, ZS01 and NB01, in the Zhoushan and Ningbo sea areas in 2014 (Figure 4). The two buoys can automatically monitor the water quality parameters with a one-hour frequency. The monitored parameters include the pH, salinity, dissolved oxygen, Chl-a, etc. Hourly data of the Chl-a concentrations collected by ZS01 and NB01 covering the period from January 1, 2015 to August 31, 2017 were selected as the experimental dataset.

The monitoring of Chl-a in Zhejiang coastal waters is quite difficult due to the high turbidity characteristics of coastal areas and the instability of data transmission. Therefore, abnormal data and missing data are occasionally received. Considering the high frequency (i.e., 1 hour) of data acquisition, a time-based linear interpolation was utilized to supplement the data, which was testified to be effective in practical applications. In addition, because of the uncertainty of the monitoring sensors in the buoys, the data quality was not completely reliable. Therefore, the original hourly data were averaged to the frequency of 12 hours to improve the data reliability. Subsequently, a total count of 1948 half-daily observations for each buoy was achieved, and these data were separated into training, validation and testing subsets. The first 1,460 (75%) and middle 300 (15%) data records were employed for training and validation, and the remaining 188 (10%) data records were utilized to test the models (Figure 5). Particularly, for the ARIMA model, the validation data were combined into training data.

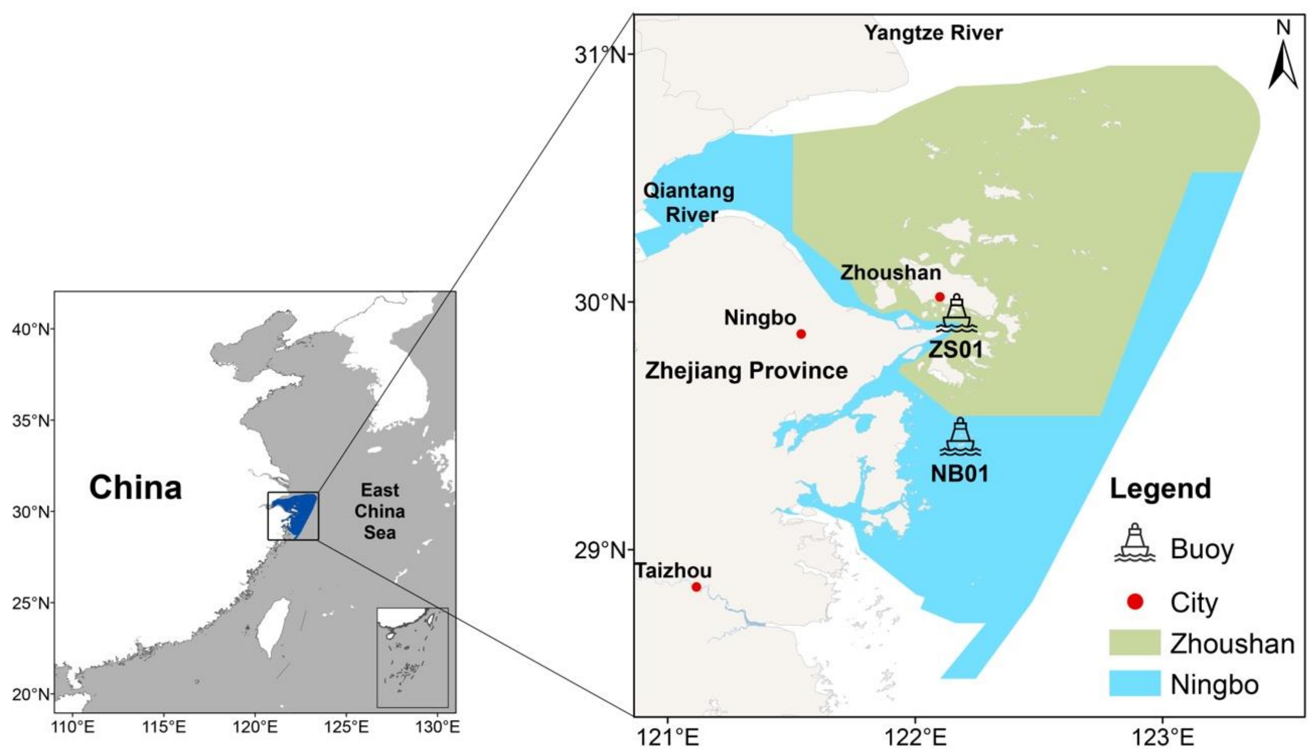

Figure 4. Study area and the locations of buoys ZS01 and NB01.

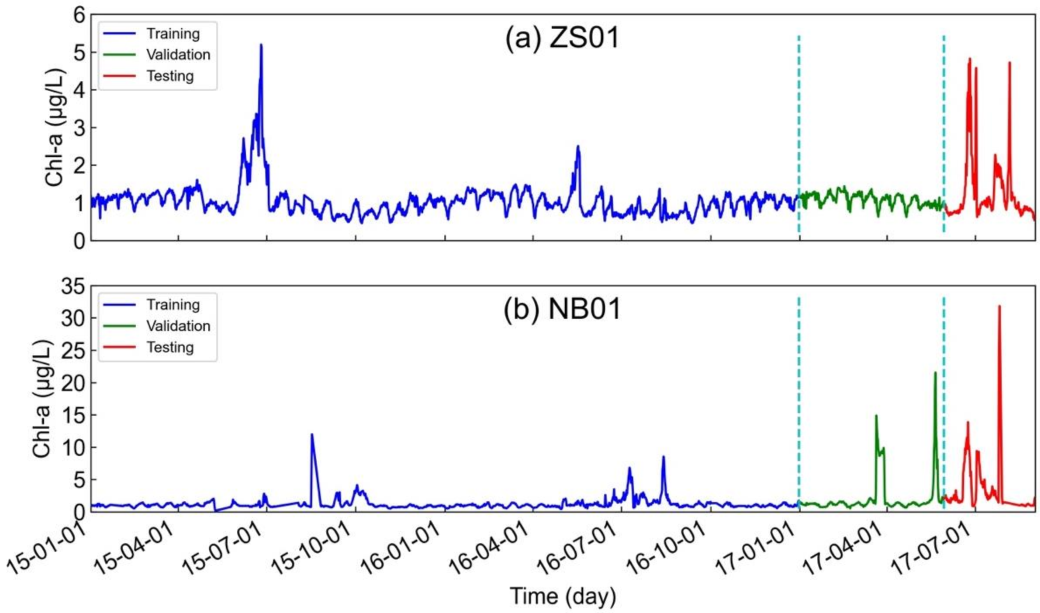

Figure 5. Temporal variations of the datasets at buoys (a) ZS01 and (b) NB01.

Table 1. Statistical Analysis Results for the Training, Validation, and Testing Data and All Datasets

	ZS01				NB01			
Statistics	All	Train	Validation	Test	All	Train	Validation	Test
Mean	1.07	1.04	1.08	1.26	1.57	1.24	2.06	3.36
Std.	0.46	0.42	0.16	0.84	2.05	1.02	2.81	4.30
Min	0.46	0.46	0.62	0.54	0.18	0.18	0.54	0.86
Max	5.20	5.20	1.44	4.82	31.83	11.98	21.53	31.83
Count	1,948	1,460	300	188	1,948	1,460	300	188
R_1	0.90	0.95	0.82	0.80	0.91	0.95	0.91	0.87
R_4	0.66	0.77	0.48	0.43	0.43	0.61	0.49	0.18
R_8	0.49	0.63	0.15	0.20	0.25	0.34	0.25	0.02
R_{16}	0.32	0.44	−0.11	0.04	0.13	0.11	0.03	−0.02

Table 1 presents the statistical analysis results of the training, validation, testing and whole Chl-a data. For each parameter, the mean, standard deviation (Std.), minimum, maximum, count, and autocorrelation coefficients of lags 1, 4, 8, and 16 (R_1, R_4, R_8, and R_{16}) were calculated. In addition, the dataset was normalized to the range [–1, 1] in the experiments.

The temporal variations of the dataset are displayed in Figure 5. Due to the influence of coastal algal blooms, the Chl-a concentrations of both buoys showed several sudden rises and formed mutation peaks during the period from May to September each year.

4. Experiment Modelling

To better indicate and forewarn the algal blooms, multi-step-ahead predictions of Chl-a are preferable. In contrast to one-step-ahead time series forecasts, multi-step-ahead predictions are generally subjected to growing uncertainty from the error accumulation. For a better comparison of the models, multi-step-ahead predictions of Chl-a were conducted. Since the sudden increase of Chl-a was rapid and generally occurred within four days, as shown in Figure 5, this article attempted to predict the Chl-a for the next 4 days based on the previous sequence, i.e., eight-step forward predictions.

Table 2. Optimal Parameters and AIC Values of the ARIMA models

ARIMA	p	d	q	AIC
ZS01	25	0	0	2,741.95
NB01	15	0	0	7,970.88

4.1. ARIMA Modelling

To fit the available Chl-a data with the ARIMA model, the autocorrelation functions and partial autocorrelation functions were utilized to determine the possible structures. The best fitted model among the many competing models was selected based on the Akaike information criterion (AIC). As shown in Table 2, the components (p, d, q) of the best ARIMA model were (25, 0, 0) and (15, 0, 0) for datasets ZS01 and NB01, respectively. The optimum AICs were 2,741.95 and 7,970.88.

4.2. ANN Modelling

Based on the experimental data, six ANN models, i.e., the MLP, ERNN, GRU, LSTM, attention-based encoder-decoder GRU (AEDGRU), and attention-based encoder-decoder LSTM (AEDLSTM), were implemented in Keras 2.0.4 with the TensorFlow 1.1.0 backend using Python language.

A three-layer MLP model, one of the most common ANN architectures, was developed to predict Chl-a using a back-propagation algorithm. The one single hidden layer network structure of the RNN models has also been demonstrated to be effective in various prediction researches (Brunelli et al., 2006; Ma et al., 2015; Tian and Pan, 2015; Zhang et al., 2017b). Therefore, the three-layer architecture was also employed to build the ERNN, LSTM and GRU models. Because the AED LSTM and AEDGRU contained encoder and decoder RNNs, they had two hidden layers. Since the dataset was normalized to the range [–1, 1], the tanh activation function was utilized in the hidden layer of the MLP and ERNN models. The activation functions of the LSTM and GRU units were sigmoid and tanh functions, and additional details are provided in Section 2.4.

The sum of square errors was adopted as the loss function and used to search the optimal model. The Adam (Kingma and Ba, 2014) algorithm was employed to train each model with a mini-batch of 100 sequences. The maximum number of training epochs was set to 5,000, and the validation dataset was fitted after each epoch to prevent over-fitting.

The neurons of the output layers of the MLP, ERNN, LSTM and GRU models were set to 1 so that the multi-step-ahead predictions of these models were obtained by an iterative forecast process. Because the outputs of the decoder of the AEDRNN models can be regarded as the subsequent serial predictions, the multi-step-ahead predictions were straightforward for the AEDRNN models by setting the neuron size of the output layer. Accordingly, the output layer's neuron size of the AEDLSTM and AEDGRU models were set to 8.

Determining the optimal sizes of the neurons in the input and hidden layers are frequently difficult; thus, a simple search strategy was conducted in this study. The optimal neuron size of input layer for each model, i.e., the size of the input time sequence, was selected from the list of [10, 20, 30, 40, 50, 60, 70, 80, 100, 150], and the neuron size of the hidden layer was optimized from the list of [3, 6, 12, 24, 48, 96, 192]. The neuron sizes of the encoder and decoder of the AEDRNN models were set to the same size.

Table 3. Optimum Architectures of the MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM Models for the ZS01 and NB01 Datasets

	ZS01			NB01		
Model	Input	Hidden	Output	Input	Hidden	Output
MLP	30	24	1	60	48	1
ERNN	60	24	1	60	48	1
GRU	60	48	1	60	48	1
LSTM	60	48	1	60	48	1
AEDGRU	60	[48, 48]	8	60	[48, 48]	8
AEDLSTM	60	[48, 48]	8	60	[48, 48]	8

Table 4. Multi-Step-Ahead Prediction Results of the ZS01 Testing Dataset

ZS01	Step 1 (12 h)			Step 2 (24 h)			Step 3 (36 h)			Step 4 (48 h)		
	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE
ARIMA	0.52	0.64	14.61	0.69	0.42	21.42	0.77	0.30	27.21	0.83	0.23	29.21
MLP	0.43	0.75	14.56	0.52	0.64	19.99	0.61	0.52	25.39	0.67	0.45	29.95
ERNN	0.41	0.78	11.67	0.53	0.63	16.49	0.56	0.58	18.25	0.58	0.55	18.60
GRU	*0.31*	*0.87*	*11.48*	*0.38*	*0.80*	*15.34*	*0.40*	*0.79*	*17.11*	0.44	0.73	20.23
LSTM	*0.30*	*0.88*	*11.83*	*0.38*	*0.80*	*15.50*	0.43	0.74	18.46	0.48	0.69	20.85
AEDGRU	**0.26**	**0.91**	**9.40**	**0.31**	**0.88**	**11.82**	**0.33**	**0.86**	**13.42**	**0.39**	**0.80**	**16.88**
AEDLSTM	*0.29*	*0.88*	*10.70*	*0.37*	*0.82*	*13.83*	*0.40*	*0.78*	*16.80*	*0.43*	*0.77*	*17.76*
	Step 5 (60 h)			Step 6 (72 h)			Step 7 (84 h)			Step 8 (96 h)		
ARIMA	0.86	0.18	30.81	0.91	0.13	34.89	0.95	0.09	37.08	0.97	0.06	38.06
MLP	0.69	0.43	33.08	0.75	0.36	37.03	0.79	0.31	39.51	0.83	0.28	41.83
ERNN	0.58	0.55	19.73	0.59	0.55	25.51	0.61	0.52	28.80	0.63	0.50	29.64
GRU	0.45	0.72	21.90	0.47	0.69	24.26	0.56	0.57	29.92	0.61	0.50	33.58
LSTM	0.52	0.62	22.66	0.55	0.58	24.58	0.60	0.49	27.01	0.63	0.45	29.38
AEDGRU	*0.44*	*0.74*	*20.18*	*0.48*	*0.68*	*22.91*	*0.49*	*0.67*	*23.72*	*0.50*	*0.67*	*25.03*
AEDLSTM	**0.42**	**0.82**	**17.94**	**0.45**	**0.76**	**20.52**	**0.44**	**0.79**	**19.84**	**0.47**	**0.75**	**21.14**

Note that the model with the best capability was marked in bold with a light orange background, and the one with the second-best performance was marked in italics with a light grey background.

Table 5. Multi-Step-Ahead Prediction Results of the NB01 Testing Dataset

NB01	Step 1 (12 h)			Step 2 (24 h)			Step 3 (36 h)			Step 4 (48 h)		
	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE
ARIMA	2.31	0.76	19.03	3.58	0.46	25.88	4.28	0.23	38.10	4.75	0.11	49.60
MLP	1.60	0.88	22.26	2.78	0.62	33.87	3.45	0.43	45.07	4.03	0.25	54.32
ERNN	1.42	0.89	16.60	2.68	0.65	27.02	3.78	0.39	42.06	5.03	0.18	60.07
GRU	**1.00**	**0.95**	**9.35**	**1.63**	**0.86**	**15.39**	*2.07*	*0.78*	*22.39*	*2.47*	*0.68*	*29.34*
LSTM	*1.05*	*0.94*	*11.25*	*1.75*	*0.85*	*15.98*	*2.15*	*0.79*	*21.41*	*2.49*	*0.73*	*26.39*
AEDGRU	1.41	0.90	13.38	2.02	0.79	18.69	2.26	0.74	23.79	2.50	0.69	30.89
AEDLSTM	1.14	0.94	12.02	*1.68*	*0.89*	*16.85*	**1.92**	**0.87**	**20.64**	**2.14**	**0.83**	**24.88**
	Step 5 (60 h)			Step 6 (72 h)			Step 7 (84 h)			Step 8 (96 h)		
ARIMA	5.05	0.04	63.19	5.15	0.01	72.35	5.15	0.00	78.97	5.13	0.00	78.60
MLP	4.43	0.14	62.47	4.70	0.08	69.51	4.79	0.06	70.26	4.75	0.05	67.11
ERNN	5.86	0.08	84.41	6.37	0.04	106.99	6.97	0.01	131.85	7.64	0.00	155.52
GRU	2.78	0.60	36.75	3.03	0.52	44.08	3.25	0.45	49.69	3.44	0.39	52.66
LSTM	2.74	0.69	31.74	2.92	0.67	36.37	3.05	0.67	38.95	3.17	0.67	39.60
AEDGRU	*2.59*	*0.69*	*31.02*	*2.65*	*0.69*	*33.57*	*2.55*	*0.76*	*32.23*	*2.62*	*0.73*	*33.21*
AEDLSTM	**2.25**	**0.84**	**27.68**	**2.31**	**0.86**	**30.50**	**2.35**	**0.85**	**32.02**	**2.38**	**0.89**	**30.99**

Note that the model with the best capability was marked in bold with a light orange background, and the one with the second-best performance was marked in italics with a light grey background.

Therefore, 70 models were developed for each model and compared using the two statistical measures R^2 and RMSE. The optimum architectures of the six ANN models for the datasets of ZS01 and NB01 are reported in Table 3.

Through simple sensitivity analysis, the capacities and impacts of different neural network structures on the final prediction results can be learned. For various input neuron sizes, we found that the optimal neuron size of input layer was 60 for almost all models except the MLP model of ZS01 as shown in Table 3. In other words, the best input temporal sequence of Chl-a prediction for both ZS01 and NB01 was mainly 60 / 2 = 30 days, which demonstrated that the temporal correlation period of Chl-a that can be efficiently fitted was about 30 days in our study area. Moreover, with regard to the selection of hidden neuron size, when the neuron size of the hidden layer gradually increased to the optimal size (i.e., the MLP and ERNN models of ZS01 were 24, and the other models were 48), the model performance of both training and validation datasets kept rising, which indicated that the models were still under-fitting, but when the neuron size continued to increase and exceeded the optimal size, the model performance of the training dataset probably still heightened slightly while that of the validation dataset rapidly dropped, showing that the models began to overfit.

The difference between the Spearman correlation from NLR and that from LR in Figure 5 again has the dash and dot-dash lines lying above the horizontal axis in (b) indicating NLR generally outperforming LR for non-outliers, and lying below the horizontal axis in (a) indicating NLR generally underperforming LR for outliers.

5. Results and Discussion

Tables 4 and 5 demonstrate the multi-step-ahead prediction performance of different models using the test dataset for both ZS01 and NB01 buoys. The model with the best capability was marked in bold with a light orange background, and the one with the second-best performance was marked in italics with a light grey background.

The performance of the robust RNN models (LSTM, GRU, AEDLSTM and AEDGRU) was considerably better than that of the ARIMA, MLP and ERNN models according to the RMSE, R^2 and MAPE statistics, which was probably because the robust RNNs are capable of memorizing long-term dependencies (Hochreiter and Schmidhuber, 1997; Gers, 2001; Cho et al., 2014). For buoys ZS01 and NB01, the robust RNNs presented accuracy improvements of at least 30% over the ARIMA and MLP models and at least 20% over the ERNN model in most cases. Considering that robust RNN models were more effecttive than the other models, we focus on the performance comparison between the robust RNNs as well as between the remaining models in the subsequent discussion. For better comparisons, we divided the forecast steps into short-term predictions (steps 1 ~ 4) and middle-term predictions (steps 5 ~ 8).

Among the ARIMA, MLP and ERNN models, in the case of the ZS01 dataset, the ERNN was the best model for both short-term and middle-term predictions, and its performance for middle-term predictions was even comparable to that of the LSTM and GRU models. However, for the NB01 dataset, the short-term predictions capabilities were still maintained, but the performance for middle-term predictions drastically declined and became even worse than that of the ARIMA and MLP models. This finding revealed that the ERNN model was relatively unstable and did not always learn successfully, which was consistent with the study conducted by Ma et al. (2015). In addition, the MLP achieved better results than the ARIMA for both short-term and middle-term predictions. Overall, the ERNN model was slightly superior to the MLP model, and both of these models were stronger than the ARIMA for modelling our dataset.

Detailed comparisons between the robust RNN models were also conducted. With respect to the ZS01 dataset, the AEDGRU model achieved the best results for short-term predictions, since its RMSE, R^2 and MAPE results were superior to those of the GRU, LSTM, and AEDLSTM models. Also, we noticed that the performance of the AEDLSTM model was superior to that of the GRU and LSTM models, although the improvement was not obvious, especially in the one-step-ahead and two-step-ahead predictions.

With regard to the NB01 dataset, notable results were observed for the short-term predictions. The performance of the GRU and LSTM for one-step-ahead and two-step-ahead predictions was even better than that of the AEDGRU and AEDLSTM, and the performance was still superior to the AEDGRU although inferior to that of the AEDLSTM for three-step-ahead and four-step-ahead predictions.

However, in terms of middle-term predictions, the performance of the AEDLSTM and AEDGRU models was considerably superior to that of the LSTM and GRU models for both the ZS01 and NB01 datasets, which indicated that the AEDRNN models were more stable and accurate than the LSTM and GRU models for long time-series prediction.

When comparing the LSTM and GRU models, the GRU's performance was superior to that of the LSTM in most cases. Specifically, for short-term predictions, the GRU exceeded the LSTM for both the ZS01 and NB01 datasets overall. While for middle-term predictions, the GRU was basically superior to the

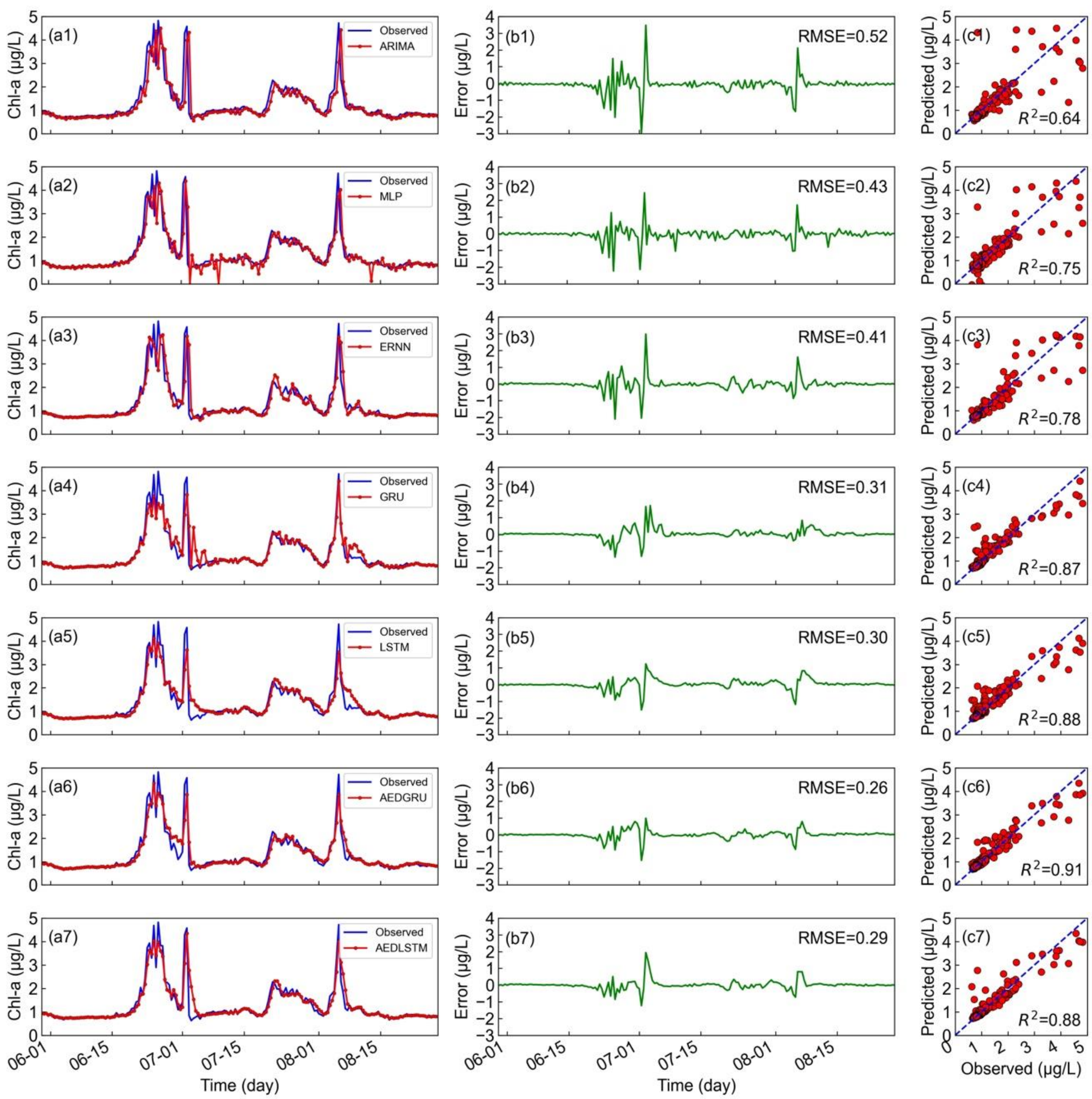

Figure 6. Performance comparison of the ARIMA, MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM models for the one-step-ahead prediction using the ZS01 dataset.

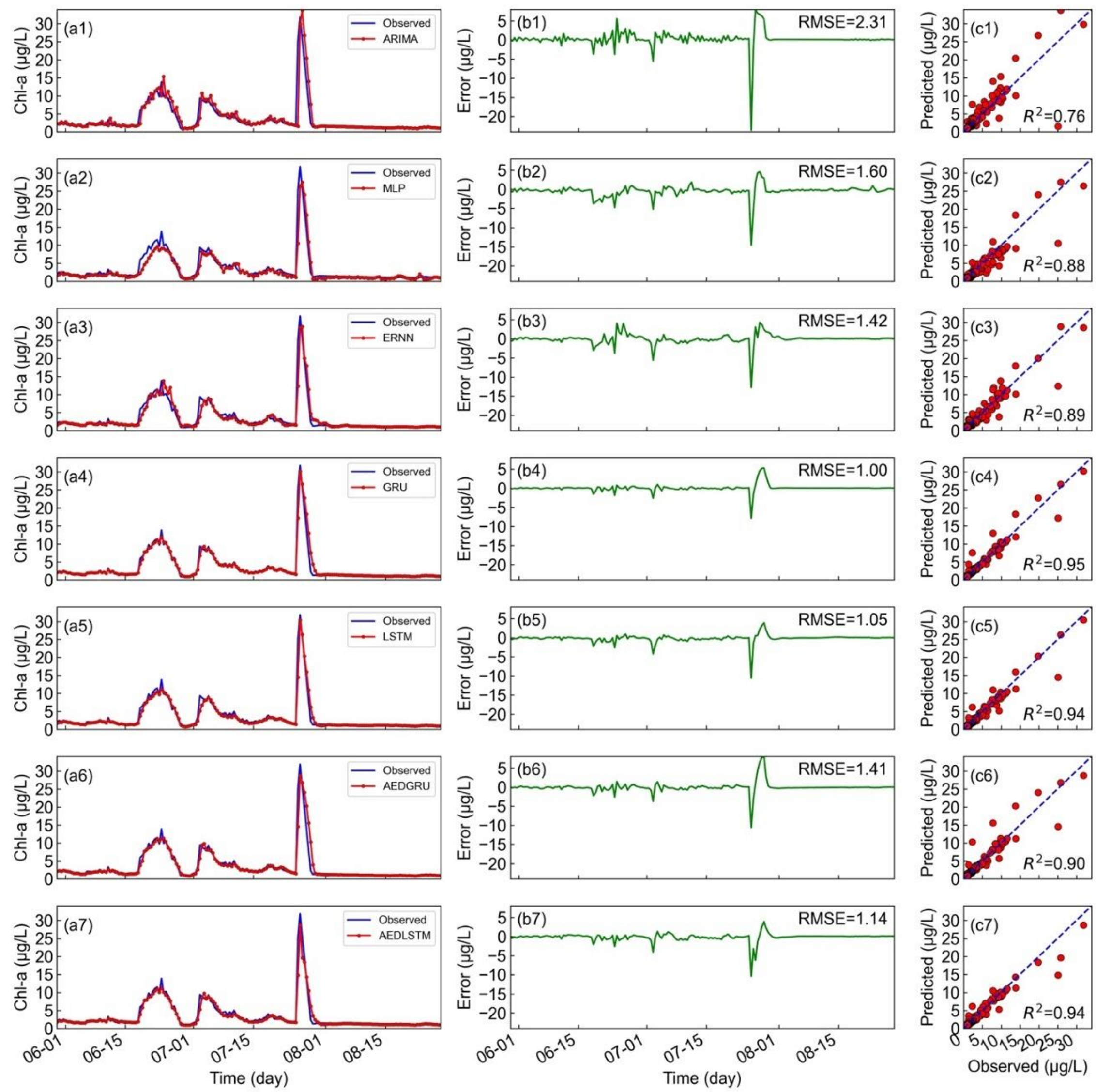

Figure 7. Performance comparison of the ARIMA, MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM models for the one-step-ahead prediction using the NB01 dataset.

LSTM for the ZS01 dataset but inferior to the LSTM for the NB01 dataset. The GRU required even less training time benefited from its simplified architecture (Cho et al., 2014); thus, we concluded that the GRU was better than the LSTM for Chl-a prediction in this study.

When comparing the AEDLSTM and AEDGRU models, their short-term prediction performances were comparable, but the AEDLSTM significantly outperformed the AEDGRU for middle-term predictions according to the RMSE, R^2, and MAPE results, which demonstrated that the AEDLSTM was more robust than the AEDGRU.

In addition, it can be observed that some statistical indicators of the later-step prediction were even better than the earlier-step prediction for several models, such as the R^2 of eight-step-ahead prediction of AEDLSTM for the NB01 dataset. Also, we found that such phenomenon mainly occurred in the middle-term predictions of the AEDRNN models, probably because the most relevant encoder states were selected and integrated through the attention-based temporal feature combination, which effectively reduced the accumulated errors and improved the prediction accuracy of the later-steps. On the other hand, predicting extreme value is often difficult, especially for the middle-term prediction. However, extremum data usually has a large prediction bias and has a considerable impact on the statistical indicators, which probably leads to a lower accuracy. Therefore, the uncertainty and contingency of the extremum prediction in the middle-term prediction also increased the probability that the statistical indicators of later-step prediction surpassed the previous prediction.

To further test the forecasting performance in a more intuitive manner, the time series of the observed and predicted Chl-a using the ARIMA, MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM models for the test period of both the ZS01 and NB01 buoys were plotted (Figures 6 ~ 9). Furthermore, the prediction errors and the scatter plots of the predictions against the observations using the models were also plotted. Because too many figures are required to show all the predicte-dresults, the results of the one-step-ahead and eight-step-ahead predictions are presented.

Figures 6 and 7 show that all models presented good consistency between the real and predicted values for the one-step-ahead predictions. But according to the prediction error figures, the robust RNNs were better able to fit the sudden changes of Chl-a, whereas the other models showed a delayed prediction trend. This result was likely because of the insufficient learning ability of the traditional ANN and ARIMA models (Elman, 1990; Le et al., 2017). Moreover, the results of the robust RNNs were closer to the 1:1 line in the scatter figures than the other models.

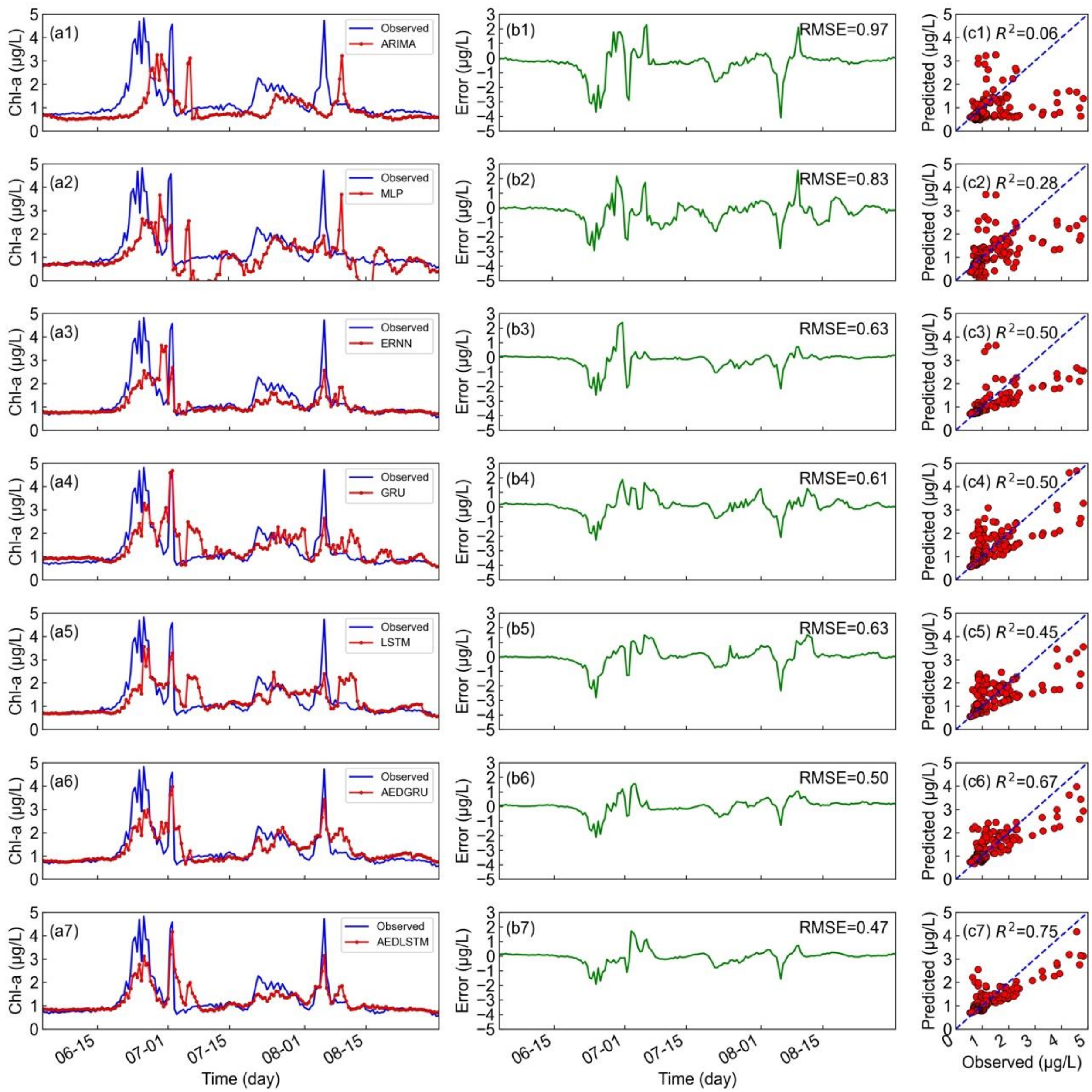

Figure 8. Performance comparison of the ARIMA, MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM models for the eight-step-ahead predictions using the ZS01 dataset

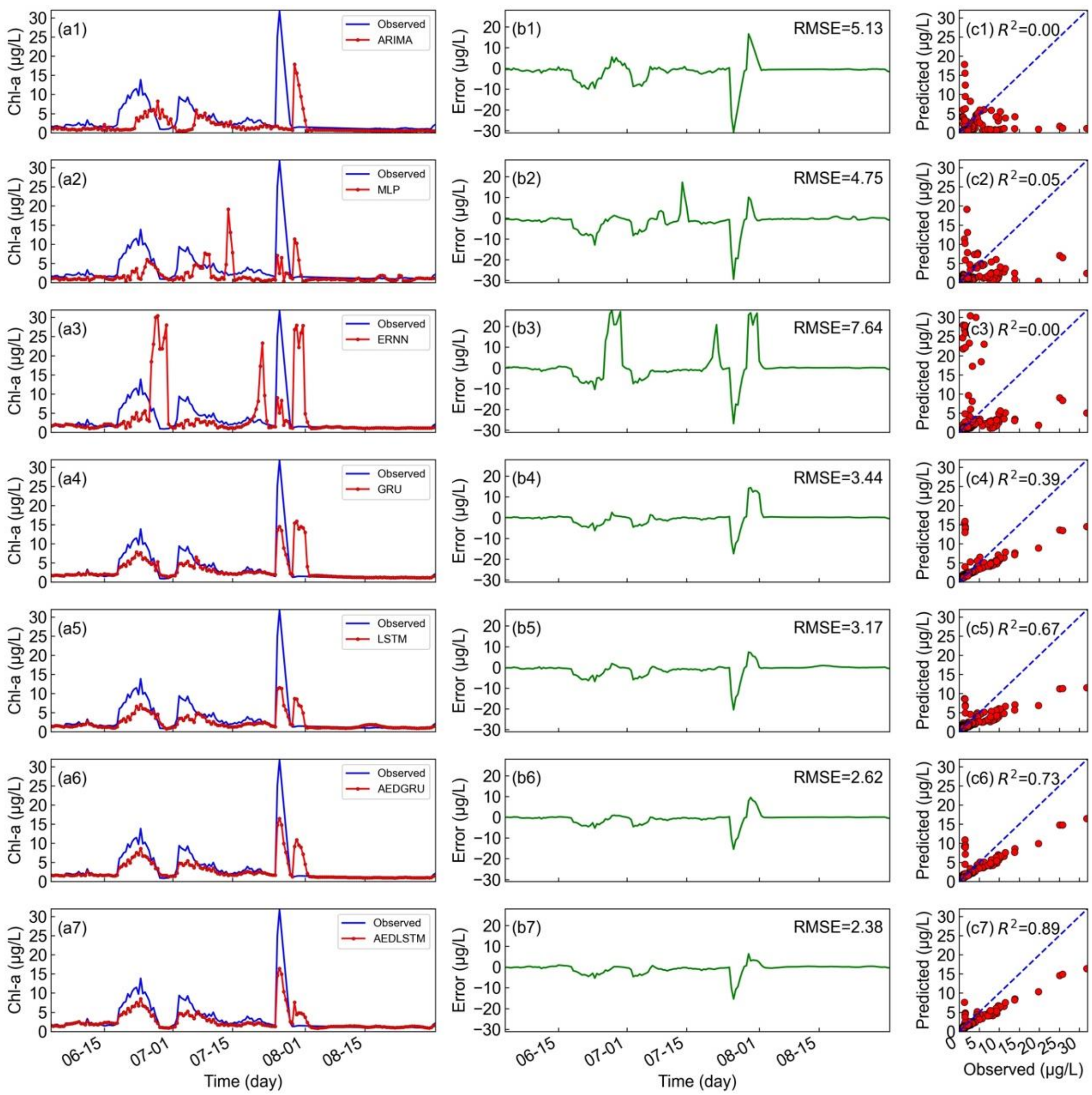

Figure 9. Performance comparison of the ARIMA, MLP, ERNN, GRU, LSTM, AEDGRU and AEDLSTM models for the eight-step-ahead prediction using the NB01 dataset.

The performance differences between these models for the eight-step-ahead prediction were more obvious as shown in Figures 8 and 9. The ARIMA, MLP, ERNN, GRU and LSTM models were not able to provide accurate predictions or well-capture the tendencies, although the GRU and LSTM models had better results than the ARIMA, MLP and ERNN models. Notably, the AEDGRU and AEDLSTM models obtained more accurate forecasting results and more consistent trends than the GRU and LSTM models, due to the capability of encoder-decoder mechanism and the consideration of attention-based temporal feature combination (Bahdanau et al., 2014).

Based on the above analysis, several useful findings were observed:

(1) By exploiting the long-term dependency memorizing capability, the robust RNN models (GRU, LSTM, AEDGRU and AEDLSTM) achieved considerably higher performance than the ARIMA, MLP and ERNN models, which confirmed the applicability of the robust RNN models for Chl-a predictions.

(2) The AEDRNN models (AEDGRU and AEDLSTM) outperformed the GRU and LSTM models, although the performance of the GRU and LSTM for short-term predictions was comparable to that of the AEDRNN models. Considering that the middle-term predictions of Chl-a were important for generating accurate inferences on algal blooms (Wang et al., 2015), the AEDRNN models were better than the GRU and LSTM models. However, if only short-term predictions were needed in certain areas, the GRU and LSTM models could be the better choices since they are more efficient.

(3) The performance of the GRU model was superior to that of the LSTM model in most cases of our experiments. Be-

cause the GRU model was more efficient with its simplified architecture, we concluded that the GRU model was better than the LSTM model for Chl-a predictions.

(4) Although the AEDLSTM and AEDGRU models were comparable for short-term predictions, the AEDLSTM achieved significantly better results than the AEDGRU for middle-term predictions. In summary, the AEDLSTM model was considered to be the best model for Chl-a predictions in this study.

6. Conclusions

To accurately predict Chl-a in the coastal areas, this paper introduced the LSTM and GRU models to capture the long-term dependencies, and based on the LSTM and GRU units, two simplified AEDRNN models were also developed for time series forecasting. To validate the effectiveness of the robust RNN models, the Chl-a data from the ZS01 and NB01 buoys in the Zhejiang coastal area covering the period from January 1, 2015 to August 31, 2017 were collected. The first 75% of the data and the middle 15% of the data were used for training and validation, and the last 10% of the data was utilized to test the models. In addition, the ARIMA, MLP, and ERNN models were compared with the robust RNN models based on the same dataset.

The results provided several helpful conclusions. (1) The robust RNN models outperformed other algorithms in terms of stability and accuracy, which confirmed the applicability of the robust RNN models for Chl-a predictions. (2) By exploiting the capabilities of encoder-decoder mechanism and attention-based feature combination, the AEDRNN models exceeded the GRU and LSTM models, especially for long time-series predictions. (3) The GRU was a better choice than the LSTM because the GRU obtained a superior performance and higher efficiency than the LSTM in most cases of the experiments. (4) The AEDLSTM was the most robust and suitable model for Chl-a predictions in terms of its much higher performance than all other models.

In addition, some general guidance can be concluded for the time-series forecasting of other phenomena. Specifically, the ARIMA and MLP models are not quite accurate for prediction, but their computation efficiency is superior to other models, so they are suitable for simple and fast predictions. The ERNN model has higher accuracy than the ARIMA and MLP models, but is unstable so that it doesn't always learn successfully. For greater accuracy and stability, the robust RNN is a better choice. Among them, the accuracy of short-term prediction of GRU and LSTM is comparable to the AEDRNN models. Considering the higher efficiency, GRU and LSTM are the better choice for short-term forecasting, while AEDRNN will be the best for middle-term forecasting.

Future work should focus on including external covariate information in the robust RNN models, which implies that various water quality parameters can be used as additional inputs. In addition, the performance of these models for long-term predictions should be further studied. Another interesting direction of future research is to combine these models into a hybrid architecture to improve the learning ability.

References

Ahmad, S., Khan, I.H. and Parida, B.P. (2001). Performance of stochastic approaches for forecasting river water quality. *Water Res.*, 35(18), 4261-4266. https://doi.org/10.1016/S0043-1354(01)00167-1

Zaytar, M.A. and El Amrani, C. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *Int. J. Comput. Appl.*, 143(11), 7-11. https://doi.org/10.5120/ijca2016910497

Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computer Science*. https://doi.org/10.48550/arXiv.14 09.0473 (preprint)

Bao, W., Yue, J. and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), e0180944. https://doi.org/0.1371/journal.pone.0180944

Box, G.E. and Jenkins, G.M. (1976). *Time series analysis: forecasting and control*. John Wiley & Sons, pp 97-105. ISBN: 978-1-118-67502-1

Brunelli, U., Piazza, V. and Pignato, L. (2006). A forecast model to predict the next day's maximum hourly SO_2 in the site of Priolo (Siracusa). *WIT Trans. Ecol. Environ.*, 99,10. https://doi.org/10.2495/RAV060571

Chen, Q., Guan, T., Yun, L., Li, R. and Recknagel, F. (2015). Online forecasting chlorophyll a concentrations by an auto-regressive integrated moving average model: Feasibilities and potentials. *Harmful Algae*, 43, 58-65. https://doi.org/10.1016/j.hal.2015.01.002

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Computer Science*. https://doi.org/10.48550/arXiv.1406.1078 (preprint)

Cinar, Y.G., Mirisaee, H., Goswami, P., Gaussier, E., Aït-Bachir, A. and Strijov, V. (2017). Position-based content attention for time series forecasting with sequence-to-sequence RNNs. *Neural Information Processing. 24th International Conference, Guangzhou, V-24*, 533-544.

Elman, J.L. (1990). Finding structure in time. *Cognitive Sci.*, 14(2), 179-211. https://doi.org/10.1207/s15516709cog1402_1

Fan, X., Zhang, H., Leung, C. and Miao, C. (2016). Comparative study of machine learning algorithms for activity recognition with data sequence in home-like environment. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 168-173. https://doi.org/10.1109/MFI.2016.7849484

Faruk, D.Ö. (2010). A hybrid neural network and ARIMA model for water quality time series prediction. *Eng. Appl. Artif. Intel.*, 23(4), 586-594. https://doi.org/10.1016/j.engappai.2009.09.015

Fu, R., Zhang, Z. and Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE*, 324-328. https://doi.org/10.1109/YAC.2016.7804912

Gers, F. (2001). *Long short-term memory in recurrent neural networks*. PH.D. Dissertation, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Harada, M., Tominaga, T., Hiramatsu, K. and Marui, A. (2013). Real-time prediction of chlorophyll-a time series in a eutrophic agricultural reservoir in a coastal zone using recurrent neural networks with periodic chaos neurons. *Irrig. Drain.*, 62, 36-43. https://doi.org/10.1002/ird.1757

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *Computer Science*. https://doi.org/10.48550/arXiv.1412.6980 (preprint)

Le, J. A., El-Askary, H.M., Allali, M. and Struppa, D.C. (2017). Application of recurrent neural networks for drought projections in California. *Atmos. Res.*, 188, 100-106. https://doi.org/10.1016/j.atm osres.2017.01.002

Lee, J.H., Huang, Y., Dickman, M. and Jayawardena, A.W. (2003). Neural network modelling of coastal algal blooms. *Ecol. Model*., 159(2-3), 179-201. https://doi.org/10.1016/S0304-3800(02)00281-8

Ma, X., Tao, Z., Wang, Y., Yu, H. and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transport. Res. C-Emer*., 54, 187-197. https://doi.org/10.1016/j.trc.2015.03.014

Maknickienė, N. and Maknickas, A. (2012). Application of neural network for forecasting of exchange rates and forex trading. *The 7th International Scientific Conference Business and Management*, 122-127. https://doi.org/10.3846/bm.2012.017

Melesse, A.M., Krishnaswamy, J. and Zhang, K. (2008). Modeling coastal eutrophication at florida bay using neural networks. *J. Coastal Res*., 24(2B), 190-196. https://doi.org/10.2112/06-0646.1

Nicholls, K.H. (2012). Phosphorus and chlorophyll in the Bay of Quinte: A time-series/intervention analysis of 1972-2008 data. *Aquat. Ecosyst. Health*, 15(4), 421-429. https://doi.org/10.1080/14634988.2012.727720

Photphanloet, C., Treeratanajaru, W., Cooharojananone, N. and Lipikorn, R. (2016). Biochemical oxygen demand prediction for Chaophraya river using alpha-trimmed ARIMA model. *IEEE 2016 13th International Joint Conference on Computer Science and Software Engineering*, 520-525. https://doi.org/10.1109/jcsse.201 6.7748930

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G. and Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. *Computer Science*. https://doi.org/10.48550/arXiv.1704.02971 (preprint)

Rajaee, T. and Boroumand, A. (2015). Forecasting of chlorophyll-a concentrations in south san francisco bay using five different models. *Appl. Ocean Res*., 53, 208-217. https://doi.org/10.1016/j. apor.2015.09.001

Ravuri, S. and Stolcke, A. (2016). A comparative study of recurrent neural network models for lexical domain classification. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 6075-6079. https://doi.org/10.1109/ICASSP.2016.7472844

Samli, R., Sivri, N., Sevgen, S. and Kiremitci, V.Z. (2014). Applying artificial neural networks for the estimation of chlorophyll-a concentrations along the istanbul coast. *Pol. J. Environ. Stud*., 23(4), 1281-1287. https://doi.org/10.1007/BF00238283

Sutskever, I., Vinyals, O. and Le, Q.V. (2014). Sequence to sequence learning with neural networks. *Adv Neural Inf process Syst*, 4, 3104-3112. https://doi.org/10.48550/arXiv. 1409.3215

Tian, Y. and Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. *2015 IEEE International Conference on Smart City/socialcom/sustaincom*, 153-158. https://doi.org/10.1109/SmartCity.2015.63

Wang, H., Yan, X., Chen, H., Chen, C. and Guo, M. (2015). Chlorophyll-a predicting model based on dynamic neural network. *Appl. Artif. Intell*., 29(10), 962-978. https://doi.org/10.1080/08839514.2015.1097142

Wang, Y., Yang, N. and Gao, Y.B. (2016). Bootstrapped ANN for forecasting seawater chlorophyll-a around the north Pacific Rim. *Oceans, In OCEANS 2016*, Shanghai, I-1, 1-5.

Zhang, F., Sun, X., Zhou, Y., Zhao, C., Du, Z. and Liu, R. (2017). Ecosystem health assessment in coastal waters by considering spatio-temporal variations with intense anthropogenic disturbance. *Environ. Modell. Softw*., 96, 128-139. https://doi.org/10.1016/j.envsoft.2017.06.052

Zhang, G.P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175. https://doi.org/10.1016/S0925-2312(01)00702-0

Zhang, Q., Wang, H., Dong, J., Zhong, G. and Sun, X. (2017). Prediction of sea surface temperature using long short-term memory. *IEEE Geosci Remote Sens Lett*, 14(10), 1745-1749. https://doi.org/10.1109/LGRS.2017.2733548