



Article

LSTM Networks to Improve the Prediction of Harmful Algal Blooms in the West Coast of Sabah

Fatin Nadiah Yussof ¹, Normah Maan ^{1,*} and Mohd Nadzri Md Reba ^{1,2}

¹ Department of Mathematics, Faculty of Science and Technology, Universiti Teknologi Malaysia, Skudai 81310, Malaysia; fatinnadiah5@gmail.com (F.N.Y.); nadzri@utm.my (M.N.M.R.)

² Faculty of Geoinformation and Real Estate, Universiti Teknologi Malaysia, Skudai 81310, Malaysia

* Correspondence: normahmaan@utm.my

Abstract: Harmful algal bloom (HAB) events have alarmed authorities of human health that have caused severe illness and fatalities, death of marine organisms, and massive fish killings. This work aimed to perform the long short-term memory (LSTM) method and convolution neural network (CNN) method to predict the HAB events in the West Coast of Sabah. The results showed that this method could be used to predict satellite time series data in which previous studies only used vector data. This paper also could identify and predict whether there is HAB occurrence in the region. A chlorophyll *a* concentration (Chl-*a*; mg/L) variable was used as an HAB indicator, where the data were obtained from MODIS and GEBCO bathymetry. The eight-day dataset interval was from January 2003 to December 2018. The results obtained showed that the LSTM model outperformed the CNN model in terms of accuracy using RMSE and the correlation coefficient *r* as the statistical criteria.

Keywords: chlorophyll *a*; CNN; LSTM; prediction; satellite data



Citation: Yussof, F.N.; Maan, N.; Md Reba, M.N. LSTM Networks to Improve the Prediction of Harmful Algal Blooms in the West Coast of Sabah. *Int. J. Environ. Res. Public Health* **2021**, *18*, 7650. <https://doi.org/10.3390/ijerph18147650>

Academic Editor: Paul B. Tchounwou

Received: 11 June 2021

Accepted: 8 July 2021

Published: 19 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Algal bloom is a phenomenon in which phytoplankton (algae) grows excessively in a river, lake, or sea. Harmful algal bloom (HAB) is a type of harmful algal bloom that causes toxicity in humans and marine life, resulting in negative impacts on the environment, human health, and economy. Environmental impacts include the death of marine organisms. Human health impacts include health deterioration as a result of the consumption of contaminated seafood, which can lead to serious illness or mortality. Economic impacts include the wreckage of tourism attraction spots, since activities such as fishing and snorkeling cannot be carried out, as well as the decline of the aquaculture sector due to enormous fish killings. For example, on 11 February 2014, due to the HAB event in Tanjung Kupang, Malaysia, the operators reported millions of ringgits of losses, affecting the fishery sector in Malaysia. Fish stocks, like snappers, cods, seabass, and threadfins, in nine farms were wiped out during the event [1]. In Penang, the aquaculture operators reported losses estimated at around RM20 million due to fish killings in 2005–2006 [1]. The growth in the frequency of HABs has alarmed the world over the decades, and it has become a major worldwide research topic among researchers [2–8].

Previous research has identified several parameters that influence the occurrence of HABs, but all of them require favorable environmental conditions, such as nutrient concentrations [9], light availability [10], water column stratification, or changes in water temperature [11]. Traditionally, the water sampling method was used in the affected areas for lab-based cell taxonomy. However, this conventional method is labor-intensive, and limits the scope of investigation in terms of space and time [12]. This contradicts the remote sensing-based detection method, which provides greater coverage, requires less time, and, most importantly, is less expensive. The chlorophyll *a* (Chl-*a*) concentration provides an indicator of eutrophication, as it measures the concentration of algae in a water body, and hence, it is used in remote sensing-based detection methods to detect HABs. The duration

of HABs ranges from weeks to months, and they can cover up to thousands of square kilometers [13,14].

Only certain regions are covered by HAB forecasting systems, such as the National Oceanic and Atmospheric Administration (NOAA) HAB forecasting systems in the Florida region [15]. This system does not use machine learning (ML), and focuses solely on the human impact of HABs, such as respiratory problems and so on, with forecasts only lasting up to four days. HAB Observing System (HABSOS) is another system developed by NOAA [16] for HAB detection within the Gulf of Mexico.

The HAB events cause major concern due to their severe consequences to a variety of sectors. Prevention measures need to be taken for early detection for an appropriate HABs monitoring program to be constructed, and predictions need to be made to reduce the loss and damage caused by HABs [17]. From 2009 onwards, various predictive models have been successfully employed to predict HAB events, including the artificial neural network (ANN) [18–20], support vector machine (SVM) [12], random forest [13], neuro-fuzzy [21,22], and least square support vector machine (LSSVM) [23–26]. It is challenging to evaluate the HAB process, as it is a complex phenomenon with nonlinear factors. As a result, despite the outlined methods, a more advanced model is required to predict HAB events.

An advanced subfield of ML for artificial intelligence, known as deep learning (DL), has recently received considerable attention. Long short-term memory (LSTM) and convolutional neural network (CNN) are two popular models in DL [27]. LSTM is a recurrent neural network (RNN) that collects extended sequential data in the hidden memory for processing, representation, and storage. The constancy of time information is updated continually [28]. Meanwhile, the CNN's structure consists of convolutional layers in which the neurons in the layers are connected to a small region of neurons in the input data. This is followed by sliding a weight matrix, called a filter, over the input and the convolution (or dot product) computed at each point, referred to as the feature map, between the input and the filter. This architecture enables the model to learn the filter needed to recognize identified patterns in the input data [29].

HAB data are represented by the concentration of chlorophyll *a* in the water body. In 2019, chlorophyll *a* concentration in water bodies was predicted using DL models. A CNN model was used to estimate chlorophyll *a* concentration in Daechung Lake, South Korea, with an emphasis on data skewness and imbalance [30]. The researchers indicated that using log transformation and oversampling techniques improved the performance of the CNN-based prediction model, particularly for small regions. Meanwhile, LSTM was applied for the multistep-ahead prediction of chlorophyll *a* in Gongju, South Korea [31]. The results revealed that the LSTM network model achieved higher accuracy than the dense network model and batch normalization, as a regularization method aided the learning process.

The prediction is significant because it can benefit the seafood and tourism industries, as well as the stakeholders involved. The major goals of this study are to (1) investigate the capabilities of DL models, namely the LSTM and CNN models, and (2) compare the performance of the LSTM and CNN methods in predicting HAB events. Previous research has explored both the LSTM and CNN methods, but no research has compared their performances in handling satellite data. Furthermore, to the best of the authors' knowledge, this is the first time the CNN and LSTM models are used to predict HAB events on the West Coast of Sabah using satellite data.

2. Materials and Methods

2.1. Data

Chlorophyll *a* from ocean color remote sensing was used to investigate phytoplankton dynamics, which marked the presence of HAB. MODIS-Aqua level 3 chlorophyll *a* data products of the reprocessed (2018.1) version were provided by the NASA Ocean Biology Processing Group (<https://oceancolor.gsfc.nasa.gov>) (accessed on 25 June 2021), as used in this study. The eight-day data period was extracted from January 2003 to December 2018

with a 1 km coverage and monthly resolution, and bounded by 4.6° N to 7.5° N latitude and 11.3° E to 16.5° E longitude. Due to cloud cover, there were spatial gaps in the data, which were filled using data interpolating empirical orthogonal functions (DINEOFs), an EOF-based technique. The image data captured by MODIS-Aqua were loaded into MATLAB and converted into matrix numbers, in which an image represents a matrix with a size of 20×19 . The matrix numbers represent the covered region, and the maximum value in the matrix was selected to become a single datum. Our objective was to predict whether HABs occurred in the selected region. This could be accomplished by utilizing the maximum value in the matrix data for each image to identify HAB occurrence. The HAB threshold is 10 mg/L [32]. Figure 1 shows the modis pixel image for two subsequent datasets in February 2003 in which the maximum value of the pixel is highlighted on the top. Figure 2 illustrates the maximum value of chlorophyll-a concentration. It can be seen that there were several times that HABs occurred during the period from 2003 until 2018.

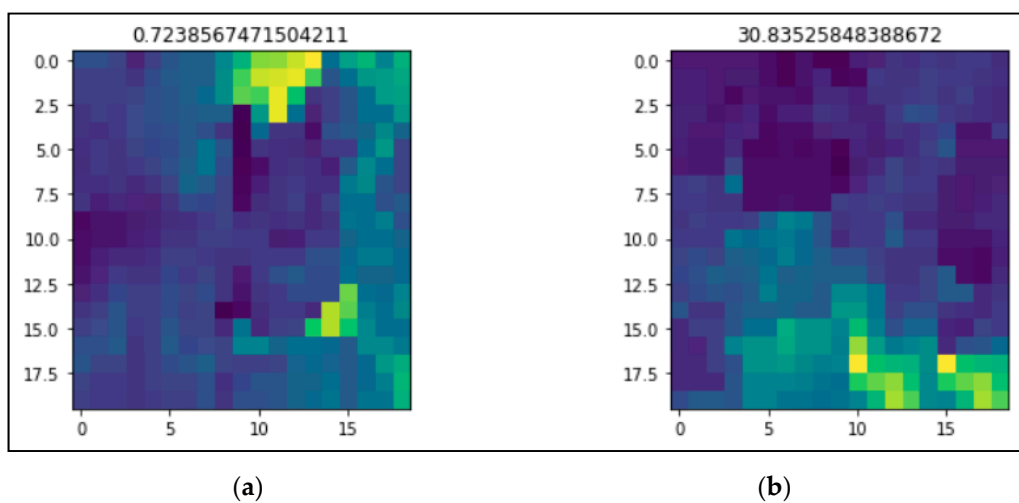


Figure 1. Plot of the modis pixel image that shows the maximum value highlighted on the top. (a) Maximum number in the modis pixel image is 0.7238567471504211 (b) Maximum number in the modis pixel image is 30.83525848388672.

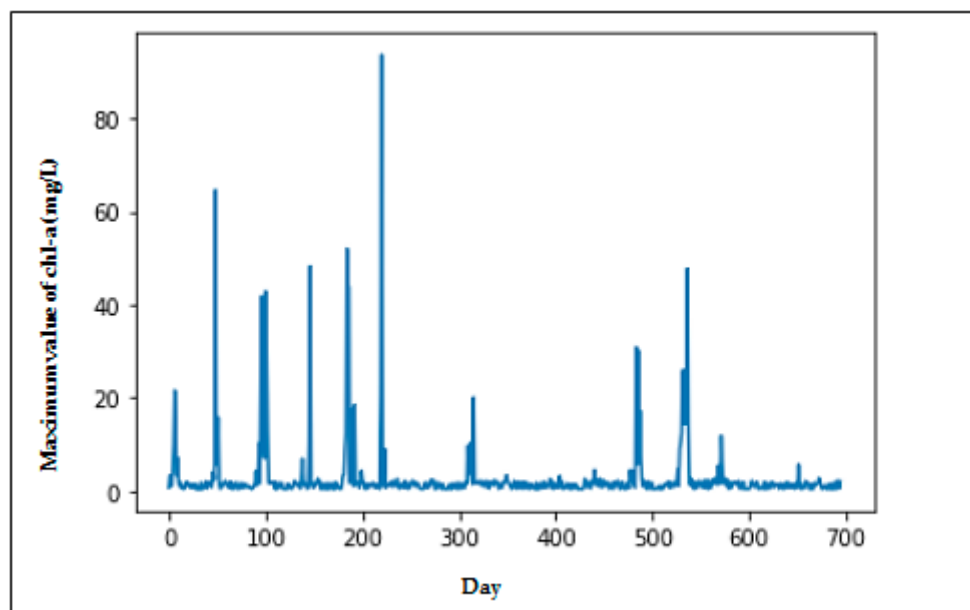


Figure 2. Plot of the maximum value of chlorophyll *a* concentrations.

The study area was Sepanggar Bay, which is located on the coastal waters of Kota Kinabalu, Sabah (Figure 3). The freshwater inflows in this region come from the Inanam River and the Menggatal River, as well as factory waste and domestic sewage. The nearby inland is the terrestrial area, while on the southern side of Sepanggar Bay are reclaimed areas and a harbor. Gaya Island acts as a protective barrier on the bay’s northwestern side, while an aquaculture project for broodstock is located on the bay’s eastern side. Due to these reasons, the productivity of the phytoplankton varies temporally and spatially.

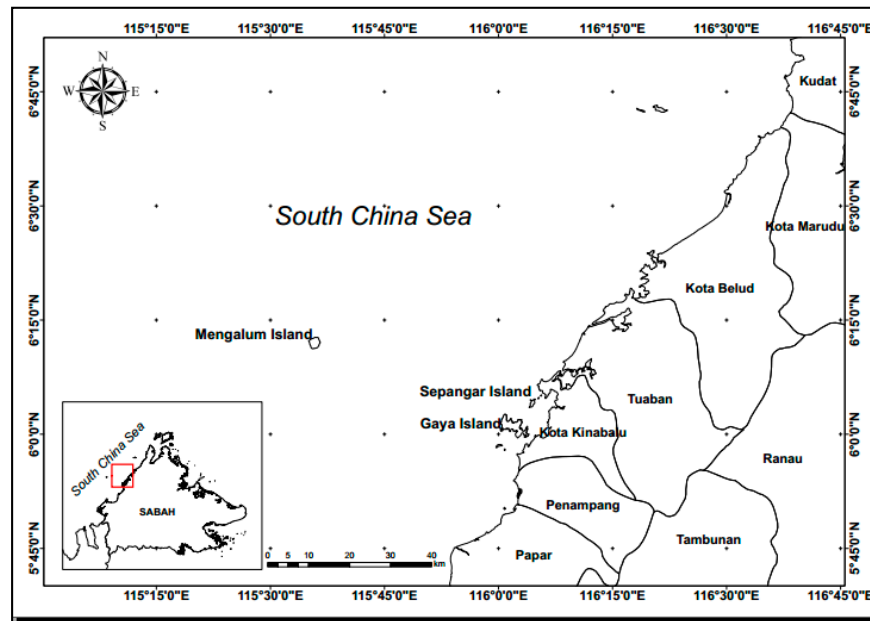


Figure 3. Location of the study area in the coastal waters of Kota Kinabalu, Sabah.

2.2. Long Short-Term Memory (LSTM)

The LSTM model, which is formulated from the recurrent neural networks (RNN) model, is capable of learning long-term dependencies, particularly changes in the hidden layer. There are three types of gates: the forget gate, input gate, and output gate. By examining the previous output and current input in the forget gate, the first sigmoid layer determines which information to discard from the cell state. A value ranging from 0 to 1 is set as a criterion for how much information to pass. A value of 0 indicates that no information is transmitted, whereas a value of 1 indicates that complete information is transmitted. The input gate determines whether the new information should be saved or discarded. Similar to the previous step, the sigmoid layer decides which value to update. Next, the tanh layer produces new results and combines the information from both layers to update the information. Finally, the output gate produces the output value by a filtered version.

$$i_t = \sigma(W_1^i \times X_t + W_h^i \times H_{t-1} + b_i) \text{ Input gate} \tag{1}$$

$$f_t = \sigma(W_1^f \times X_t + W_h^f \times H_{t-1} + b_f) \text{ Forget gate} \tag{2}$$

$$o_t = \sigma(W_1^o \times X_t + W_h^o \times H_{t-1} + b_o) \text{ Output gate} \tag{3}$$

$$\tilde{C}_t = \tanh(W_1^C \times X_t + W_h^C \times H_{t-1} + b_c) \text{ Cell entrance} \tag{4}$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-1}} \tag{5}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{6}$$

$$H_t = o_t \times \tanh(C_t) \tag{7}$$

where $W_1^i, W_1^f, W_1^o, W_1^C$ are the arrays of weights that connect X_t to the three gates, $W_h^i, W_h^f, W_h^o, W_h^C$ are the weight matrices that connect H_{t-1} to all gates and cell entrance, b_i, b_f, b_o, b_C are the bias terms of all gates and the cell entrance, σ is the logistic sigmoid function, C_t is the internal memory computed in this unit, m is a vector-only feature in every gate that obtains input from feature m of the cell vector, and H_t is the output of a hidden state derived through memory multiplication. The LSTM architecture overcomes the vanishing gradient problem in the RNN. The detailed description of the LSTM algorithm used in our prediction is described in Table 1 and the LSTM workflow for predicting HAB is shown in Figure 4, where

$C(t)$ = Internal memory of cell state

$X(t)$ = Element wise input

$H(t)$ = Output of the hidden state

$H(t-1)$ = Previous hidden state

$o(t)$ = Element of the output gate

$i(t)$ = Element of input gate

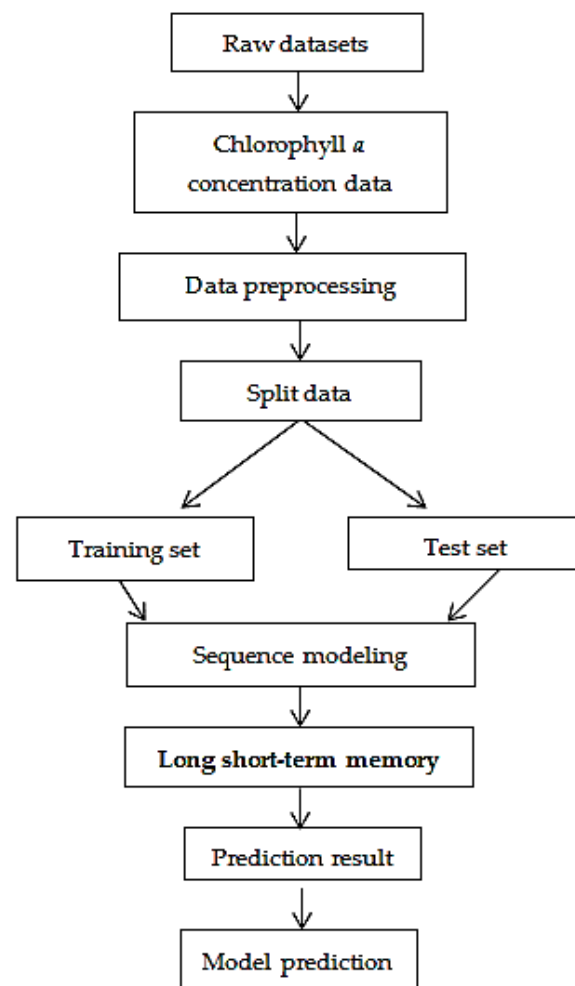


Figure 4. LSTM workflow for predicting harmful algal blooms (HABs).

Table 1. LSTM procedures.

Step	Description
1	Preprocessing of all fine particulate matter and meteorological data LSTM pre-training
2	<ul style="list-style-type: none"> Denote $X_{(t)}$ as an element-wise input, ignoring bias W is created, which represent the weight matrix to transfer the input from cell to gate vectors Elements in the gate vector are defined as m and receive information from the cell state <p>The internal memory of the cell state is denoted as C</p>
3	<p>Fine-tuning</p> <ul style="list-style-type: none"> Build a vector by applying the tanh function to the cell state <p>Apply the sigmoid function to create a filter for values of H_{t-1}.</p>
4	Obtain prediction results

2.3. Convolutional Neural Network (CNN)

The CNN is a method [33] that works well with images, particularly in image processing. Therefore, CNN is applicable for image forecasting time series. There are several layers in the CNN model, including the input layer, hidden layer, and output layer. A three-dimensional array input representing the height, weight, and number of channels is fed to a convolutional layer:

$$a^1(i, h) = (W_h^1 \times x)(i) = \sum_{j=-\infty}^{\infty} W_h^1(j) \times (i - j) \tag{8}$$

where

$$W_h^1 \in R^{1 \times k \times 1} \text{ and } a^1 \in R^{1 \times N - k + 1 \times M_1}.$$

There is only one input channel, and the output of the first layer is then passed through the nonlinear activation function $h(\cdot)$ to produce $f^{(1)} = h(a^1)$. The hidden layer, unlike other neural networks, is made up of a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer is related to the feature extraction of raw input with learnable filters [34]. Filters, also known as kernels, depict a specific feature. The CNN operates the convolution using a weight-sharing concept, which reduces the number of parameters. The weights are constantly updated during the training process [35]. The pooling layer reduces the feature dimension by producing one value out of all of the pooling window values. This reduces the input layer with a max-pooling operation, where the maximum value of the previous layer is selected [34], and aids in overcoming the problem of computational cost and overfitting [35].

The input feature map $f^{l-1} \in R^{1 \times N_{l-1} \times M_{l-1}}$ located in the hidden layer $l = 2, \dots, L$ is convolved with a set of filters $W_h^l \in R^{1 \times k \times l}, h = 1, \dots, M_1$ denoted as M_1 to produce a feature map $a^l \in R^{1 \times N_l \times M_1}$ as below [29]:

$$a^l(i, h) = (W_h^l \times f^{l-1})(i) = \sum_{j=-\infty}^{\infty} \sum_{m=1}^{M_{l-1}} W_h^l(j, m) f^{l-1}(i - j, m) \tag{9}$$

The high dimensions of the feature extracted are flattened by the fully connected layer and incorporated to get the final output:

$$l_t = \tanh(X_t \times k_t + b_t) \tag{10}$$

where l_t is the output value after convolution, \tanh is the activation function, X_t is the convolution kernel's weight, and b_t is the bias of the convolution kernel. The architecture of the CNN model is shown in Figure 5.

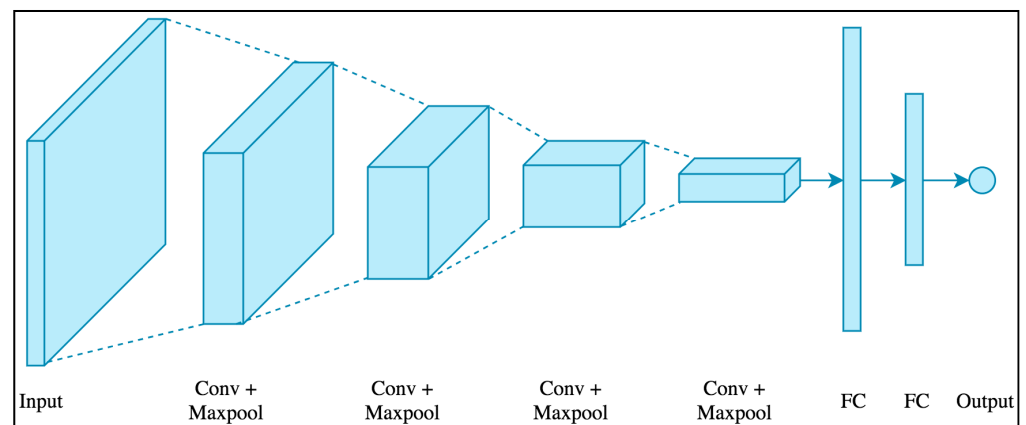


Figure 5. The CNN model architecture.

2.4. Evaluation Criteria

The accuracy of the methods is evaluated using the Pearson correlation coefficient (r) and root means square error (RMSE). In Equation (11), x represents the observed value, y represents the forecasted value, and \bar{x} and \bar{y} are the means of x and y , respectively. The formula of the Pearson correlation coefficient is as follows:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 (y - \bar{y})^2}} \quad (11)$$

In Equation (12), there are n missing data points in the test datasets, where Y_{act}^i is the actual value for the i th missing data point and Y_{est}^i is the missing data point estimated value.

$$RMSE = \sqrt{\frac{\sum_i^n (Y_{act}^i - Y_{est}^i)^2}{n}} \quad (12)$$

2.5. Training LSTM Network

One LSTM network was trained independently at a time using each one-dimensional time series corresponding to a pixel of the stacked images in the time axis. The whole satellite image time series with 640 time steps from 1 January 2003 to 31 December 2018 was divided into two partitions for data preparation: 70% for the training stage and 30% for the testing stage. The training stage was used to develop the models, while the testing stage was used to validate and compare the performance of the models in the training phase. Chl-a concentration (mg/L) was used as the sole model input data, since using a single variable as the input provides a higher accuracy than using many variables as inputs [36]. A normalization procedure via minimum–maximum scaling techniques was applied to the input and output variables scaled between 0 and 1 to prevent sudden gradient changes and smooth the convergence.

Several samples were obtained using the sliding time window-based method [37] to train the LSTM network from a single historical datum. For example, suppose the time window size is $l > n$ or a time series (x_1, x_2, \dots, x_n) with n time steps, then the first input subsequence for output x_{l+1} is (x_1, x_2, \dots, x_l) . A training sample is generated by sliding the time window one time step ahead at a time. The one time step ahead output x_t corresponds to the subsequence $(x_{t-1}, \dots, x_{t-l})$ in the time window. Therefore, training the LSTM network could produce samples $n - l + 1$.

2.6. Prediction Based on Trained LSTM

The LSTM can only predict one time step at a time. Thus, the output y_{t+1} uses an input sequence $(x_{t-l+1}, \dots, x_{t-1}, x_t)$ with an l step sliding time window, while the output y_{t+n} is predicted using the input sequence $(x_{t+n-l}, \dots, x_t, y_{t+1}, \dots, y_{t+n-1})$. This corresponds

to a strategy to implement the prediction of n steps iteratively using the predictive outputs of the previous steps to compose the input sequence to the next step. Hence, the input sequence $(x_{t+n-1}, \dots, x_t, y_{t+1}, \dots, y_{t+n-1})$ is used to predict the output y_{t+n} .

The input layer consisting of Chl-a concentration was fed to all three layers, including their lag times. The rectified linear unit (ReLU) is a piecewise linear function that returns output directly when positive and zero otherwise. Unit number 64 was used in the hidden layer, and a dropout layer with a rate of 0.001 was used after the LSTM layer to prevent overfitting and underfitting. Overfitting occurs when a network is overtrained and loses generality on the test dataset, whereas underfitting occurs when a network is poorly trained and the pattern remains unrecognized. The linear activation function and the fully connected layer, called “dense”, were applied with a unit of 1. Then, the model was compiled with an RMSE loss function and an adaptive gradient algorithm (AdaGrad) optimizer with a learning rate of 0.01. All of the components used were supported by functions of the Keras library in Python.

2.7. Prediction Based on the CNN

The CNN model was trained with a hidden layer, where a convolution layer (Conv1D) with a filter size of 64, a kernel size of 1, the same padding type, a ReLU activation type, and a uniform kernel initializer were used. Furthermore, a pooling layer with max pooling and a pooling size of 1 was applied with the maximum value from each neuron cluster in the previous layer. To overcome overfitting, a dropout layer with a rate of 0.001 was used after the CNN pooling layer [38]. A fully connected layer, called “dense”, was applied to the output layer, which used a linear activation function. Finally, the model was compiled with an RMSE loss function and an AdaGrad optimizer with a learning rate of 0.01. Similar to the LSTM method, all of the components used were supported by functions of the Keras library in Python.

3. Results

The results are summarized in Table 2. In the training and testing phase, the LSTM model, which had an $r = 0.338386$ and $RMSE = 3.402142$ mg/L, outperformed the CNN model, which had an $r = 0.111790$ and $RMSE = 4.361724$ mg/L. These findings are backed by previous research [38,39] that claimed that the LSTM model outperformed the CNN model.

Table 2. Comparison of the prediction accuracy of the long short-term memory (LSTM) model with the convolution neural network (CNN) model.

Model	RMSE	r
LSTM	3.402142	0.338385
CNN	4.361724	0.111790

Figures 6 and 7 illustrate the loss function plots of the LSTM and CNN models, respectively. In Figure 6, the graph of the loss function plots configuration illustrates that overfitting and underfitting were avoided in the LSTM model, since the curve of validation loss was always below the curve of training loss. The curve of the training loss function in the LSTM graph shows a steady improvement in the prediction quality throughout the training process. Meanwhile, the validation loss curve shows no further improvement after a small number of epochs. In contrast, the CNN loss function graph in Figure 7 shows that overfitting occurred when the number of iterations reached its maximum. This exemplifies that the CNN was incapable of capturing data with a long-term dependency.

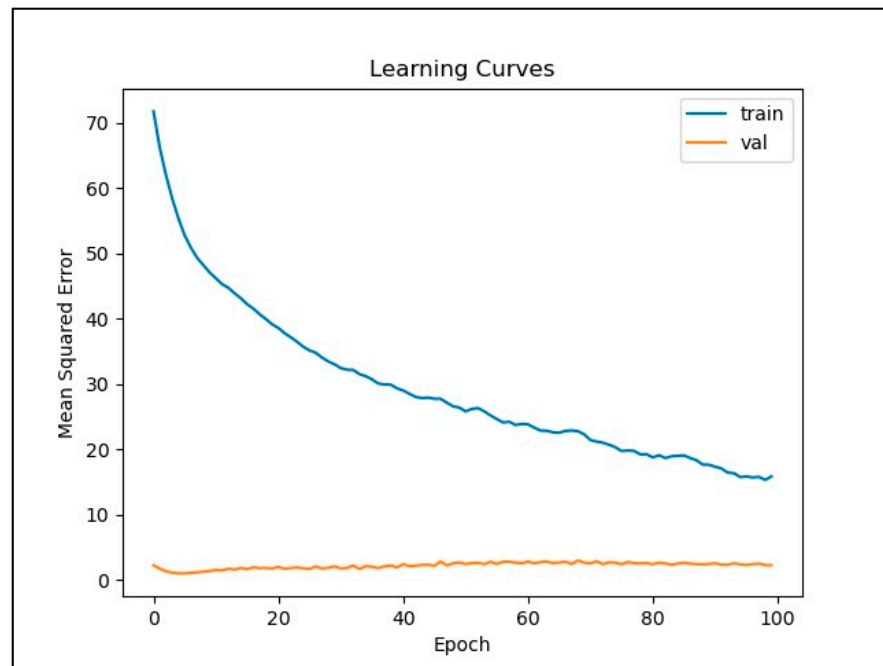


Figure 6. Loss function plots for the LSTM model.

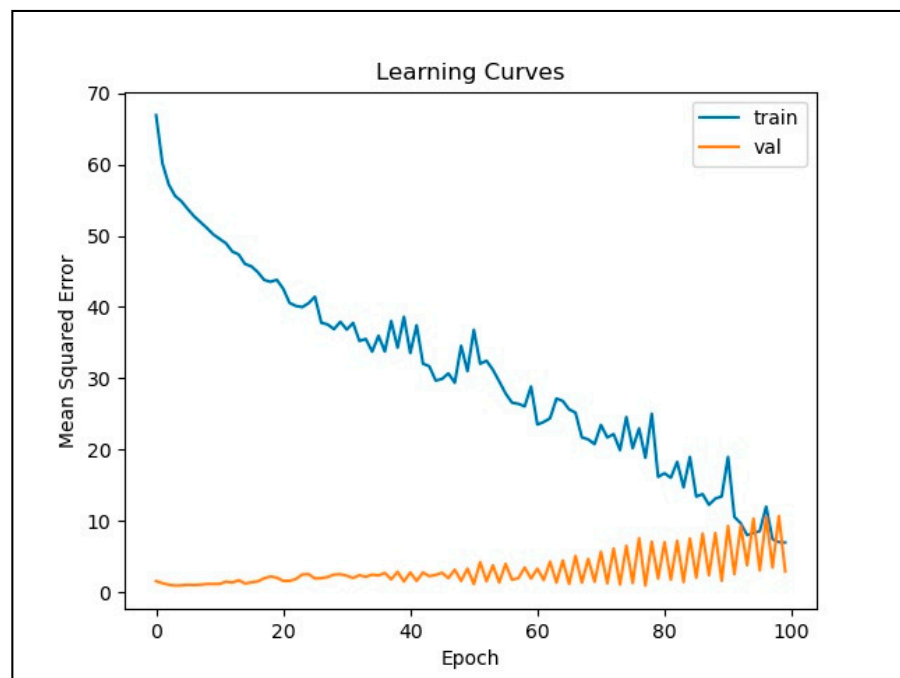


Figure 7. Loss function plots for the CNN model.

Figures 8 and 9 show the graphs of predicted and actual values in predicting HABs, respectively. The blue line is the actual data, while the red line is the predicted data. In this study, chlorophyll-a data were predicted eight days in advance. During the training, it could be seen that an HAB occurred within the period in which the value of the concentration of chlorophyll a was greater than 10 mg/L.

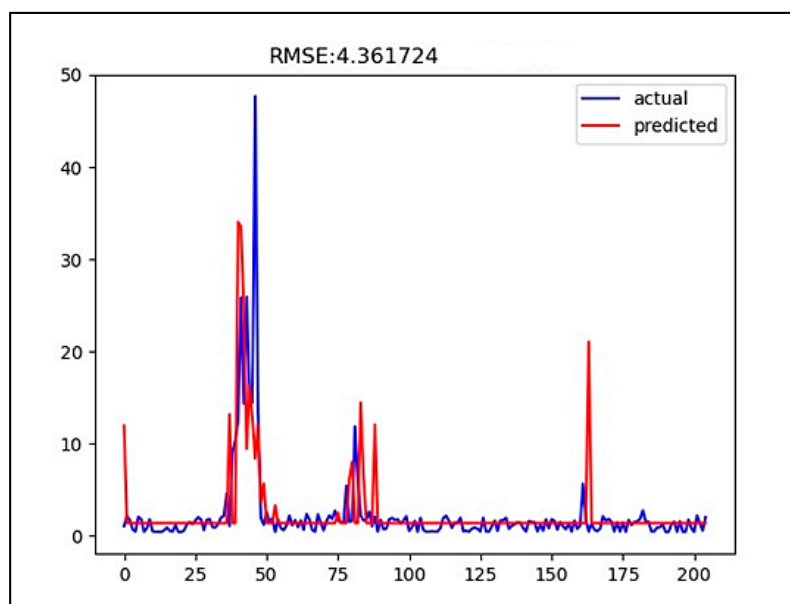


Figure 8. Comparison of predicted and actual values in predicting HABs using the CNN method.

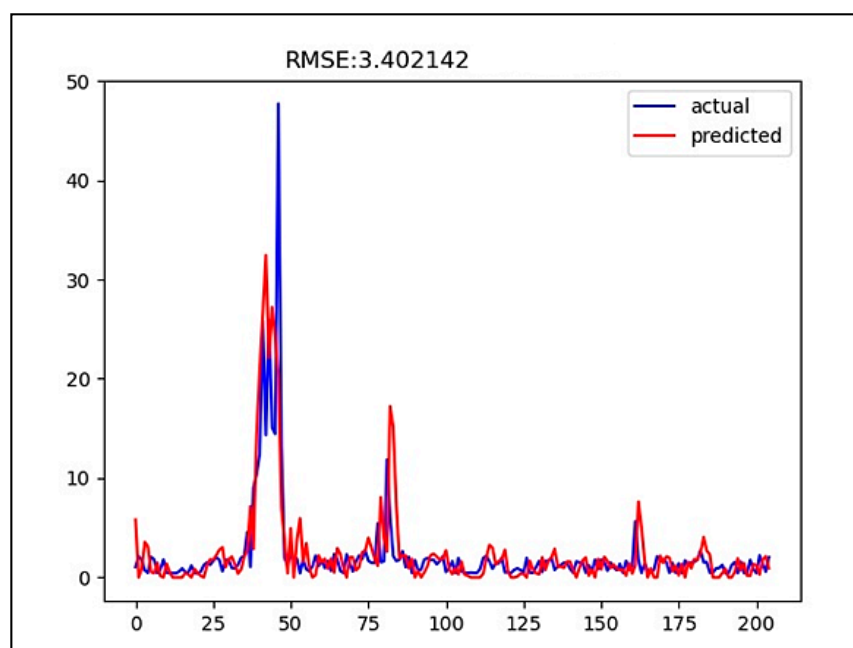


Figure 9. Comparison of predicted and actual values in predicting HABs using the LSTM method.

Figure 10 shows the scatter plots of both models, demonstrating that the LSTM model had a higher r and was well-described by linear regression ($y = 0.6229x + 1.2440$). This indicated that the LSTM model was more accurate than the CNN model, with $y = 0.3939x + 1.4066$. Both methods had a positive correlation coefficient r , indicating that, as x increased by one point, y increased by the coefficient value.

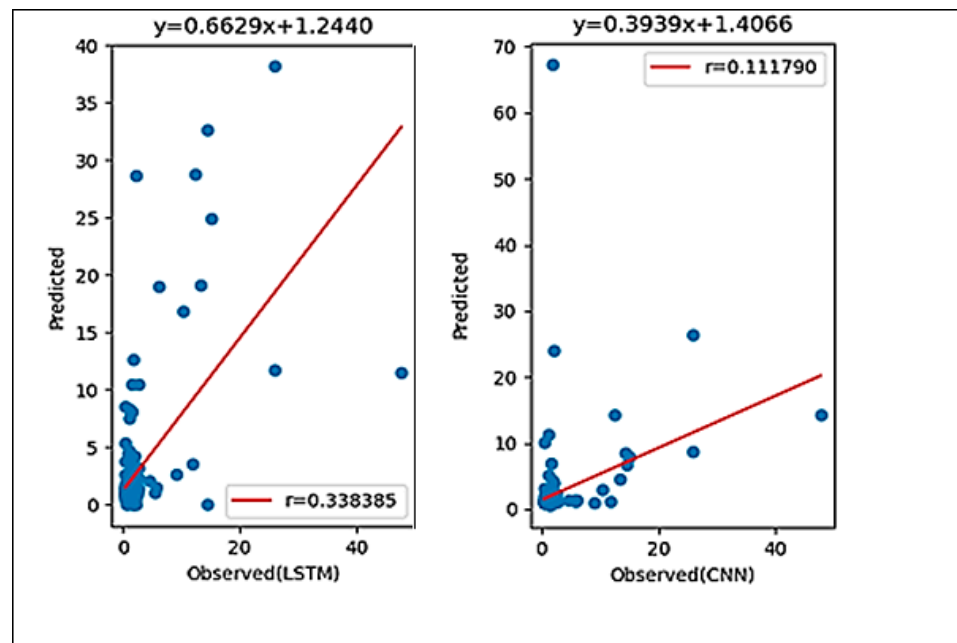


Figure 10. Plots of the observed and predicted Chlorophyll *a* concentration using the LSTM and CNN models.

4. Discussion

From the results, the LSTM method outperformed the CNN method in terms of the accuracy of r and the RMSE. These findings are backed by previous research [38,39] that claimed that the LSTM model outperforms the CNN model. This is due to the CNN model's inability to capture long-term dependencies, while the LSTM model can effectively capture long temporal dependencies. LSTM model algorithms were designed to solve the problem of information loss for long-term memory in existing RNN models, which is known to improve predictions by transferring information on previous data as the amount of data grows [23].

However, the r value is insufficient in demonstrating that the LSTM method is the best at predicting HAB events. This is due to the missing data problem, since there is the limitation of using satellite data where there is always clouds cover. The result is improved, since the missing data problems were solved using the DINEOF method [40] instead of omitting it [23]. Therefore, the purpose of this study is not to propose a new forecasting methodology, but rather to aid in the discovery of relevant approaches for predicting HAB events.

The significance of this study is to show that the LSTM model could be employed to predict HABs, since it has the ability to handle long-term dependency data. Since HABs occur annually in the West Coast of Sabah, this prediction helps to give early signals to the authorities and the region community on the occurrence of HABs. Therefore, several preventive measures can be done to cater to the adverse effects of HABs. For example, associated authorities, like the Ministry of Health, could give early warnings to the public against eating shellfish during that period to prevent the ingestion of toxic seafood that could affect human health.

It was discovered that DL models have greater flexibility in learning nonlinearity, particularly in handling large datasets. Many previous studies have demonstrated the capabilities of the CNN and LSTM models in solving short-term predictions [23,39]. In this study, the performance of the LSTM method demonstrates the capability to improve the short-term prediction accuracy model for algal blooms. The predictions eight days in advance would help to implement several preventive measures for algal bloom mitigation.

5. Conclusions

The LSTM and CNN models were used to predict HAB events on the West Coast of Sabah based on satellite data and chlorophyll *a* concentration. The data were collected from 1 January 2003 to 31 December 2018, and were divided into two partitions: 70% for training and 30% for testing. Statistical criteria, such as the RMSE and correlation coefficient *r*, were used to evaluate the testing period performance. The results showed that the LSTM model outperformed the CNN model across all statistical criteria, since the LSTM method can learn long-term dependencies, which the CNN method cannot. However, the correlation coefficient *r* of the LSTM model is still low and not very strong. For future work, studies on improving the value of *r* using other appropriate forecasting methods shall be conducted, such as using a hybrid model that combines two methods to improve the efficiency of the model. In addition, it is also suggested to study how to input the matrix data into the prediction models without having to transform into vector data.

There are some drawbacks using the LSTM model, where different random weight initialization affects this method. Instead, the model prefers small weight initialization. Not only that, but the LSTM is also prone to overfitting, and the dropout algorithm is difficult to apply to curb this issue. In addition, there are also some limitations in using satellite time series data in which there are always missing data, since there is cloud cover. Therefore, it becomes a limiting factor to obtain an accurate prediction of HABs even when the data have been corrected. Satellite data also could not identify the species blooming in that time. In other words, HABs could be predicted without knowing the species involved. In this regard, it is recommended to explore the capability of other DL models for medium- and long-term prediction in future research, since longer-term prediction is more challenging, because it requires more pertinent historical input data than short-term prediction [32].

Author Contributions: Conceptualization, N.M., F.N.Y.; Data curation, M.N.M.R.; Formal Analysis, F.N.Y.; Methodology, F.N.Y.; Software, F.N.Y.; Supervision, N.M.; Validation, F.N.Y., N.M.; Writing—original draft preparation, F.N.Y.; Writing—review and editing, F.N.Y., N.M. and M.N.M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Higher Education, Malaysia, under Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UTM/02/10), grant number R.J130000.7854.5F220.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. This data can be found here: [<https://oceancolor.gsfc.nasa.gov>] accessed on 15 July 2021.

Acknowledgments: This research was supported by the Ministry of Higher Education, Malaysia, under Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UTM/02/10), grant number R.J130000.7854.5F220. The authors are also thankful to Universiti Teknologi Malaysia for providing the facilities in this research.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Lim, P.T.; Gires, U.; Leaw, C.P. Harmful Algal Blooms in Malaysia. *Sains Malays.* **2012**, *41*, 1509–1515.
2. Blondeau-Patissier, D.; Gower, J.; Dekker, A.; Phinn, S.R.; Brando, V.E. A review of ocean color remote sensing methods and statistical techniques for the detection, mapping and analysis of phytoplankton blooms in coastal and open oceans. *Prog. Oceanogr.* **2014**, *123*, 123–144. [[CrossRef](#)]
3. Stumpf, R.P.; Culver, M.E.; Tester, P.A.; Tomlinson, M.; Kirkpatrick, G.J.; Pederson, B.A.; Truby, E.; Ransibrahmanakul, V.; Soracco, M. Monitoring karenia brevis blooms in the gulf of mexico using satellite ocean color imagery and other data. *Harmful Algae* **2003**, *2*, 147–160. [[CrossRef](#)]
4. Tomlinson, M.C.; Stumpf, R.P.; Ransibrahmanakul, V.; Truby, E.W.; Kirkpatrick, G.J.; Pederson, B.A.; Vargo, G.A.; Heil, C.A. Evaluation of the use of seawifs imagery for detecting karenia brevis harmful algal blooms in the eastern gulf of mexico. *Remote Sens. Environ.* **2004**, *91*, 293–303. [[CrossRef](#)]

5. Gokaraju, B.; Durbha, S.S.; King, R.L.; Younan, N.H. A machine learning based spatio-temporal data mining approach for detection of harmful algal blooms in the gulf of mexico. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2011**, *4*, 710–720. [[CrossRef](#)]
6. Gokaraju, B.; Durbha, S.S.; King, R.L.; Younan, N.H. Ensemble methodology using multistage learning for improved detection of harmful algal blooms. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 827–831. [[CrossRef](#)]
7. Song, W.; Dola, J.M.; Cline, D.; Xiong, G. Learning-based algal bloom event recognition for oceanographic decision support system using remote sensing data. *Remote Sens.* **2015**, *7*, 13564–13585. [[CrossRef](#)]
8. Lee, G.; Bae, J.; Lee, S.; Jang, M.; Park, H. Monthly chlorophyll—A prediction using neuro-genetic algorithm for water quality management in Lakes. *Desalination Water Treat.* **2016**, *57*, 26783–26791. [[CrossRef](#)]
9. Santoleri, R.; Banzon, V.; Marullo, S.; Napolitano, E.; D’Ortenzio, F.; Evans, R. Year-to-year variability of the phytoplankton bloom in the southern adriatic sea (1998–2000): Sea-viewing wide field-of-view sensor observations and modeling study. *J. Geophys. Res.* **2003**, *108*, 8122. [[CrossRef](#)]
10. Gohin, F.; Lampert, L.; Guillaud, J.-F.; Herbland, A.; Nézan, E. Satellite and in situ observations of a late winter phytoplankton bloom, in the Northern Bay of Biscay. *Cont. Shelf Res.* **2003**, *23*, 1117–1141. [[CrossRef](#)]
11. Thomas, A.C.; Townsend, D.W.; Weatherbee, R. Satellite-measured phytoplankton variability in the Gulf of Maine. *Cont. Shelf Res.* **2003**, *23*, 971–989. [[CrossRef](#)]
12. Craig, S.E.; Lohrenz, S.E.; Lee, Z.P.; Mahoney, K.L.; Kirkpatrick, G.J.; Schofield, O.M.; Steward, R.G. Use of hyperspectral remote sensing reflectance for detection and assessment of the harmful alga, *karenia brevis*. *Appl. Opt.* **2006**, *45*, 5414–5425. [[CrossRef](#)] [[PubMed](#)]
13. Zhang, F.; Wang, Y.; Cao, M.; Sun, X.; Du, Z.; Liu, R.; Ye, X. Deep-learning-based approach for prediction of algal blooms. *Sustainability* **2016**, *8*, 1060. [[CrossRef](#)]
14. Park, Y.; Cho, K.H.; Park, J.; Cha, S.M.; Kim, J.H. Development of early-warning protocol for predicting chlorophyll-a concentration using machine learning models in freshwater and estuarine reservoirs, Korea. *Sci. Total. Environ.* **2015**, *502*, 31–41. [[CrossRef](#)] [[PubMed](#)]
15. Li, X.; Sha, J.; Wang, Z.-L. Application of feature selection and regression models for chlorophyll-a prediction in a shallow lake. *Environ. Sci. Pollut. Res.* **2018**, *25*, 19488–19498. [[CrossRef](#)]
16. Yajima, H.; Derot, J. Application of the Random Forest model for chlorophyll-a forecasts in fresh and brackish water bodies in Japan, using multivariate long-term databases. *J. Hydroinform.* **2018**, *20*, 206–220. [[CrossRef](#)]
17. McClain, C.R. A decade of satellite ocean color observations. *Annu. Rev. Mar. Sci.* **2009**, *1*, 19–42. [[CrossRef](#)]
18. Aghel, B.; Rezaei, A.; Mohadesi, M. Modeling and prediction of water quality parameters using a hybrid particle swarm optimization–neural fuzzy approach. *Int. J. Environ. Sci. Technol.* **2019**, *16*, 4823–4832. [[CrossRef](#)]
19. Zhu, S.; Hadzima-Nyarko, M.; Gao, A.; Wang, F.; Wu, J.; Wu, S. Two hybrid data-driven models for modeling water-air temperature relationship in rivers. *Environ. Sci. Pollut. Res.* **2019**, *26*, 12622–12630. [[CrossRef](#)]
20. Fijani, E.; Barzegar, R.; Deo, R.; Tziritis, E.; Konstantinos, S. Design and implementation of a hybrid model based on two-layer decomposition method coupled with extreme learning machines to support real-time environmental monitoring. *Sci. Total. Environ.* **2019**, *648*, 839–853. [[CrossRef](#)]
21. Najafzadeh, M.; Ghaemi, A. Prediction of the five-day biochemical oxygen demand and chemical oxygen demand in natural streams using machine learning methods. *Environ. Monit. Assess.* **2019**, *191*, 380. [[CrossRef](#)]
22. Lee, S.; Lee, D. Improved prediction of harmful algal blooms in four Major South Korea’s Rivers using deep learning models. *Int. J. Environ. Res. Public Health* **2018**, *15*, 1322. [[CrossRef](#)] [[PubMed](#)]
23. Kong, Y.L.; Huang, Q.; Wang, C.; Chen, J.; He, D. Long short-term memory neural networks for online disturbance detection in satellite image time series. *Remote Sens.* **2018**, *10*, 452. [[CrossRef](#)]
24. Liu, P.; Wang, J.; Sangaiah, A.K.; Xie, Y.; Yin, X. Analysis and prediction of water quality using LSTM deep neural networks in IoT environment. *Sustainability* **2018**, *11*, 2058. [[CrossRef](#)]
25. Xayasouk, T.; Lee, H.; Lee, G. Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models. *Sustainability* **2020**, *12*, 2570. [[CrossRef](#)]
26. Kumar, A.; Islam, T.; Sekimoto, Y.; Mattmann, C.; Wilson, B. Convcast: An embedded convolutional LSTM based architecture for precipitation nowcasting using satellite data. *PLoS ONE* **2020**, *15*, e0230114. [[CrossRef](#)]
27. Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **2019**, *182*, 72–81. [[CrossRef](#)]
28. Borovykh, A.; Bohte, S.; Oosterlee, C.W. Conditional time series forecasting with convolutional neural networks. *arXiv* **2017**, arXiv:1703.04691.
29. Choi, J.; Kim, J.; Won, J.; Min, O. Modelling chlorophyll-a concentration using deep neural networks considering extreme data imbalance and skewness. In Proceedings of the 21st International Conference on Advanced Communication Technology (ICACT), Pyeong-Chang, Kwangwoon Do, Korea, 17–20 February 2019; pp. 631–634. [[CrossRef](#)]
30. Cho, H.; Choi, U.J.; Park, H. Deep learning application to time series prediction of daily chlorophyll-a concentration. *WIT Trans. Ecol. Environ.* **2018**, *215*, 157–163.
31. Li, X.; Peng, L.; Yao, X.; Cui, S.; Hu, Y.; You, C.; Chi, T. Long short term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environ. Pollut.* **2017**, *231*, 997–1004. [[CrossRef](#)]

32. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
33. Zuo, R.; Xiong, Y.; Wang, J.; Carranza, E.J.M. Deep learning and its application in geochemical mapping. *Earth Sci Rev.* **2019**, *192*, 1–14. [[CrossRef](#)]
34. Hoseinzade, E.; Haratizadeh, S. CNN pred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [[CrossRef](#)]
35. Sameen, M.I.; Pradhan, B.; Shafri, H.Z.M.; Hamid, H.B. Applications of deep learning in severity prediction of traffic accidents. In Proceedings of the Global Civil Engineering Conference, Kuala Lumpur, Malaysia, 25–28 July 2017; Springer: Singapore, 2017; pp. 793–808.
36. Xiao, X.; He, J.; Huang, H.; Miller, T.R.; Christakos, G.; Reichwaldt, E.S.; Ghadouani, A.; Lin, S.; Xu, X.; Shi, J. A novel single-parameter approach for forecasting algal blooms. *Water Res.* **2017**, *108*, 222–231. [[CrossRef](#)] [[PubMed](#)]
37. Cheng, M.; Xu, Q.; Lv, J.; Liu, W.; Li, Q.; Wang, J. MS-LSTM: A multi-scale LSTM model for BGP anomaly detection. In Proceedings of the 2016 IEEE 24th International Conference on the Network Protocols (ICNP), Singapore, 8–11 November 2016; pp. 1–6, 8–11.
38. Gu, Y.; Lu, W.; Qin, L.; Li, M.; Shao, Z. Short-term prediction of lane-level traffic speeds: A fusion deep learning model. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 1–16. [[CrossRef](#)]
39. Barzegar, R.; Aalami, M.T.; Adamowski, J. Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. *Stoch. Environ. Res. Risk Assess.* **2020**, *34*, 1–19. [[CrossRef](#)]
40. Yussof, F.N.B.M.; Maan, N.B.; Reba, M.N.B.M. Reconstruction of Chlorophyll-a Data by Using DINEOF Approach in Sepanggar Bay, Malaysia. *Comput. Sci.* **2021**, *16*, 345–356.