Spring 5-1-2021

Application of Machine Learning Techniques to Forecast Harmful Algal Blooms in Gulf of Mexico

Bala Tripura Sundari Yerrapothu

APPLICATION OF MACHINE LEARNING TECHNIQUES TO FORECAST HARMFUL ALGAL BLOOMS IN GULF OF MEXICO

by

Bala Tripura Sundari Yerrapothu

A Thesis
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Computing Sciences and Computer Engineering
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

Approved by:

Dr. Bikramjit Banerjee, Committee Chair
Dr. Diana Bernstein, Co-Chair
Dr. Kala Marapareddy

May 2021

ABSTRACT

The Harmful Algal Blooms (HABs) forecast is crucial for the mitigation of health hazards and to inform actions for the protection of ecosystems and fisheries in the Gulf of Mexico (GoM). For the sake of simplicity of our application we assume ocean color satellite imagery from the National Oceanic and Atmospheric Administration as a proxy for HABs.

In this study we use a deep neural network trained on the 2-Dimensional time series proxy data to provide a forecast of the HABs' manifestations in the GoM. Our approach analyzes between both spatial and temporal features simultaneously. In addition, the network also helps to fill in the gaps of the time series data along the way. We use Long Short Term Memory (LSTM) layers to learn the underlying trends in the time series data and Convolutional layers to decode the spatial trends in the 2-Dimensional gridded data.

Our unique contribution is an iterative, bidirectional training scheme, where we train two models: for forward and backward prediction. The intention is that if there is a functional dependence within the data in the forward time direction, then such a dependence may also exist in the backward time direction, which may be leveraged for predictions to fill the gaps in the data. We train each model to predict the next data point in their respective time-direction, based on an LSTM recurrence over the "lookback" data points. Since there are missing cells in the grid within each data point, we use a custom loss function that ignores prediction errors on missing cells. Thus the loss function critiques the models based on known cells alone, while the models act with (forward/backward) predictions that are spatiotemporally consistent across both missing and visible cells, thus updating the input training data, and consequently changing the object of critique. This actor-critic training scheme progresses iteratively, leading to the iterative improvement of the models/actors.

Several models are developed with varying combinations of convolutional layers and max pooling layers to enable the model to learn the spatial and temporal trends within the month-long training data. The most effective model performs reasonably well with prediction of chlorophyll intensities.

ACKNOWLEDGMENTS

I express my prolific and wholehearted thanks to my research advisor Dr. Bikramjit Banerjee, Professor of Computer Science, University of Southern Mississippi for his valuable guidance and continuous encouragement to complete my thesis research work. My indebted thanks are specially for his valued suggestions and support at each and every stage of my research work. His methodology, patience, long vision, and advice in research enabled me to learn a lot. He was and remains my best good example for a researcher, mentor, and instructor. I will forever be grateful to him.

I am greatly indebted to my research supervisor Dr. Diana Bernstein, Assistant Research Professor, Department of Marine Science, University of Southern Mississippi for supporting me during these 2 years. Diana is the one you will love instantly and never forget once you meet her. She's one of the coolest advisors and the smartest professor I know. I wish that I could be as energetic, lively, and enthusiastic as Diana. Diana was the motivation behind why I chose to go to seek after a profession in research. Her love and enthusiasm for teaching is contagious.

I express my profuse and heartfelt thanks to Dr. Ramakalavathi, Assistant Professor, Department of Computer Science, University of Southern Mississippi for her precious suggestions and encouraging me always and for being the strength of support whenever needed that helped me to the core to improve my focus on the research work. I thank her for fulfilling all administrative formalities in the completion of my research.

I express my significant sentiment of appreciation to my beloved Parents, Brother, Sister and my friends, whose support enabled me to concentrate on my research and complete it successfully. Finally, I place my guidance at the feet of the ALMIGHTY.

A part of this work was supported by grant "Monitoring 2019 Bonnet Carré Spillway Impacts" AN#2000006464 from Mississippi Department of Marine Resources, awarded to Dr. Bernstein.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

LIST OF ABBREVIATIONS

HAB	-	Harmful Algal Bloom
ML	-	Machine Learning
MODIS	-	Moderate Resolution Imaging Spectroradiometer
NOAA	-	National Oceanic and Atmospheric Administration
HAB-OFS	-	Harmful Algal Bloom Operational Forecast System
SVM	-	Support Vector Machine
ELM	-	Extreme Learning Machine
LSTM	-	Long Short Time Memory
OLS	-	Ordinary Least Square
GEBCO	-	General Bathymetric Chart of the Oceans
CNN	-	Convolution Neural Network
RF	-	Random Forest
VIIRS	-	Visible Infrared Imaging Radiometer Suite
OGCM	-	Ocean General Circulation Model
RMSE	-	Root Mean Square Error

Chapter 1

INTRODUCTION

Repeated openings of the Bonnet Carré Spillway cause large ecological and economic impacts to Mississippi Sound, the consequences were never imagined when it was built 91 years ago. Usually, in order to prevent Mississippi river flooding in New Orleans, the U.S. Army Corps of Engineers open the Bonnet Carré, but Coast residents on less populated shores of South Mississippi and Louisiana feel the fallout.

Figure 1.1: Sediment plume in lake Pontchartrain in May 2011. Adapted from Nasa.gov.

Introduction of large volumes of nutrient-rich fresh river water, from the opening of the Bonnet Carré Spillway, into the lower nutrient, estuarine Lake Pontchartrain brings the algal blooms. Figure 1.1 shows the Sediment plume in lake Pontchartrain in May 2011 because of the spillway opening [1]. Freshwater inputs from the spillway have been shown to substantially change the chemistry and ecology of the lake. Spillway openings can rapidly depress lake salinities, causing most of the lake to become fresher which can persist for several months, when seasonal weather or tropical activities introduce saltwater

from the Gulf of Mexico into the lake. Preliminary field results investigating the bloom composition identified the abundance of cyanobacterial species known to produce a variety of cyanobacterial toxins. These toxins have the potential for causing human illnesses. Previous spillway openings have been associated with toxic cyanobacteria blooms, and there is a concern that blooms could occur in shoreline areas utilized by the public, possibly exposing people and/or their pets to harmful levels of algal toxins [2]. After the spillway opening, predicting the location and intensity of blooms would enable us to take proper precautions to prevent the harmful effects of these toxins.

The Harmful Algal Blooms (HABs) prediction is crucial for preventing illness and taking action for protecting ecosystems and fisheries in the Gulf of Mexico. Lary et al. [11] in his study showed that Machine Learning (ML) has an important role in solving problems related to geosciences and remote sensing. Although the application of ML based methods to science and engineering problems have been common in the last two decades, using ML for geosciences and remote sensing problems is relatively new. A more detailed review of this study is given in Chapter 2.

This thesis is structured as follows: In Chapter 2 we discuss a few studies of forecasting the HABs without ML methods, followed by discussing the application of ML in ocean sciences. Finally, we discuss works on forecasting HABs using ML methods. In Chapter 3 we discuss a few factors that hinder the prediction of HABs, and how we have overcome these factors with the help of ML. In Chapter 4 we discuss our approach, from pre-processing the data till we get the predictions results, the model training algorithm, the different models that we have studied, and our model framework. In Chapter 5 we present our experimental results, particularly the model's accuracy, its ability to fill missing data and predict future data, and its sensitivity to the amount of training data used. We offer conclusions in Chapter 6.

Chapter 2

LITERATURE REVIEW

In this chapter, we discuss several works conducted on forecasting the HABs. We first discuss works without ML methods and then move on to works using ML methods. Initially the forecast of HABs study was highly involved in identifying the high presence of HABs location. Like Derner et al. [5] and Keeney et al. [9] studies which concentrated on identifying the HAB's presence and then used the bulletins of HAB-OFS for forecast. Briefly talking about their studies.

Derner et al. [5] presented a method to forecast HABs by pre-processing satellite images from Moderate Resolution Imaging Spectroradiometer (MODIS) Aqua imagery, NOAA Coast Watch program to highlight algal blooms. Derner et al. [5], used an algorithm to identify the high presence of chlorophyll location.The main problem that they encountered is, the identified portions are not just *Karenia brevis (K. brevis;* a dinoflagellate, associated with red tide*)* [14] but consists of blooms of other algal species also. The Harmful Algal Bloom Operational Forecast System (HAB-OFS) uses the satellite Ocean color imagery for their analysis to identify the intensity and movement by using algorithms to target the specific properties of *K. brevis* blooms in Gulf of Mexico - the relative particulate backscatter and the spectral shape characteristics in the blue-green portion of the spectrum, centered with 490 nm which employs a combination of automated processing and manual analyses of data to create decision support tools and products to mitigate HAB impacts. The comparison of analysis between the current chlorophyll anomaly product and the ensemble (chlorophyll anomaly + spectral shape 490 nm + backscatter ratio) products was performed on a similar set of images and their results were evaluated and that proved the ensemble imagery products performed better than the chlorophyll anomaly product by approx 77.5% of the time, Figure 2.1 shows the similar comparison thus by decreasing the over-prediction of bloom presence. In view of these outcomes, the ensemble product was progressed to activities and consolidated into the HAB-OFS bulletins starting in September 2015.

On the other side Keeney et al. [9] tracked the HABs' movements by its effects on the environment (using particulate trajectories and respiratory irritations in the population). In their study, the Gulf of Mexico HAB-OFS maintained by NOAA, issues the weekly bulletins that help in early recognition of *K. brevis* and assists in response efforts. User feedback and observations are used for the evaluation of forecast quality and bulletin utilization. Forecasts of transport heading and the level of respiratory irritation are given for both Florida and

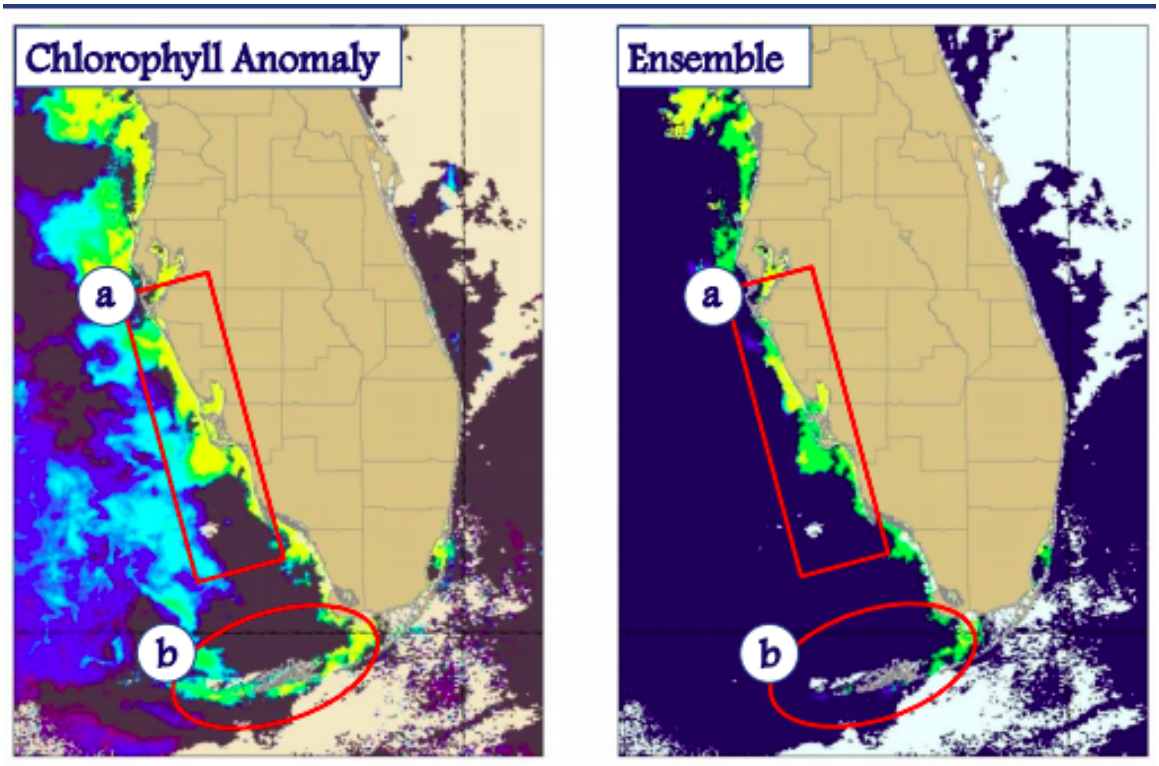

Figure 2.1: Adapted from Figure 3 Derner et al. [5]. (a) The full and backscatter ensemble products enhanced analysis along shore South West Florida Coast from approximately Tampa to Cape Romano. (b) Ensembles did not completely eliminate false positives in the Florida Bay and Florida Keys region, but HAB flags were smaller.

Texas, with the last being the most noteworthy performing. The performance of "high" respiratory irritation forecasts is considered important, in order to protect human health because the general public at that level may experience noticeable discomfort. For the areas where future prediction cannot be confirmed resulted in the data gaps. In order to improve the protocols of the HAB-OFS program for future forecasting, the results were compared against previous operational years.

Slowly after ML has started showing its progress in geosciences. The studies on forecasting the blooms using ML techniques has also started [3, 11].

Ahmad et al. [3] study summarize ML capability in oceanography. Their study says that Artificial intelligence (AI) makes it possible to integrate Machine Learning capabilities into data driven modelling systems in order to bridge the gaps and lessen demands on human experts in oceanographic research. ML algorithms have proven to be a powerful tool for analysing oceanographic and climate data with high accuracy in an efficient way and has a wide spectrum of real time applications in oceanography and Earth sciences. They also stated that the prediction of ocean weather and climate, habitat modelling and distribution,

species identification are few of the major applications of machine learning in oceanography.

Lary et al. [11] in their study presented two different examples, the first one using multivariate nonlinear nonparametric regression, and the second one using multivariate nonlinear unsupervised classification. These examples make clear scientific understanding of the real necessity of ML and how ML proved its capabilities and how it focuses on the automatic extraction of information from data by computational and statistical methods, and also in data analysis when compared with the deterministics models.

Few studies on forecasting the HABs, that include use of ML models are mentioned below.

Li et al. [13] proposed Support Vector Machines (SVMs) and neural networks as general regression to predict algal intensity by considering past observational data at specific locations. In this study, they used the biweekly data in Tolo Harbour, Hong Kong, and opted few machine learning models like backpropagation (BP) neural network, improved BP network with momentum term, Levenberg-Marquardt, Generalized regression neural network, and SVM to develop prediction models of algal blooms. In order to remove a certain randomness of the initial weights and threshold in BP networks, a genetic algorithm is additionally adopted to the BP algorithm to find the optimal initial parameters, and they compared these five ML based prediction models for chlorophyll-a concentration. When the predictions are made with different lead time, the results simulated the general trend of algal biomass reasonably. However, HAB movement in spatial dimensions has not been covered by these models so they are not suitable for precise forecasts.

Yi et al. [15] designed an intelligent model to predict chlorophyll-a concentration, which is the recognized proxy for algal activity using Etreme Learning Machine (ELM) models that use least square estimates for regression. The weights connecting the input layer to the hidden nodes are randomly assigned and are never updated. Their results say that the ELM model showed good prediction and generalization performance compared to multiple linear regression, conventional neural network with backpropagation , and adaptive neuro-fuzzy inference system.

Once the studies on prediction using Long Short Time Memory (LSTM) have started like Lee et al. [12] and Hill et al. [7], the drastic observations are seen while predicting the time series trend and predicting the regions identifying the HABs presence. Both these studies are explained below.

Lee et al. [12] conducted short-term predictions by employing regression analysis and deep learning techniques. Deep learning models like Multilayer perceptron , Recurrent Neural Network and Long Short Time Memory were used to predict chlorophyll-a concentration, a primary indicator of algal blooms. The results were compared to those from Ordinary

Least Square (OLS) regression analysis and actual data based on the root mean square error. Their conclusion is that the LSTM model showed the highest prediction rate for harmful algal blooms and all deep learning models out-performed the OLS regression analysis.

Hill et al. [7] proposed a two step prediction model using transfer learning, in which the first applies a convolutional layer to input sensor data and then the output is fed to a LSTM network to generate the final prediction. This study describes the application of machine learning techniques to develop a state-of-the-art detection and prediction system for spatiotemporal events found within remote sensing data specifically, HABs. They proposed an HAB detection system based on a ground truth historical record of HAB events, a 3-Dimensional data representation of each event from MODIS and General Bathymetric Chart of the Oceans (GEBCO) data and variety of machine learning architectures utilising state-of-the-art spatial and temporal analysis methods based on Convolutional Neural Networks (CNNs), Long Short-Term Memory components together with Random Forest and Support Vector Machine classification methods.

Their study focused on the detection of *Karenia brevis* Algae HAB events within the coastal waters of Florida over 2850 events from 2003 to 2018 (biggest ML study on HAB Data until then). The development of multimodal spatiotemporal 3-Dimensional data structures and associated novel machine learning methods gave the unique architecture for the automatic detection of environmental events. Specifically, when applied to the detection of HAB events it gives a maximum detection accuracy of 91%. A HAB forecast system was also developed where a temporal subset of each data-cube was used to predict the presence of a HAB in the future. This system was not significantly less accurate than the detection system being able to predict with 86% accuracy up to 8 days in the future.

All these studies including Derot et al. [6] and Kwon et al. [10] basically address the prediction of HAB regions and did not talk about predicting the intensity of chlorophyll present in the data. Moreover neither of these studies mentioned predicting the missing information in the data. We focus on these two problems and address them in our study.

Derot et al. [6] forecast strategy was based on pairing two machine learning models with a long-term database by creating HAB groups via a K-means (A K-means clustering algorithm tries to group similar items in the form of clusters.) model. Then they introduced different lag times in the input of a Random Forest (RF) model, using a sliding window. They used the high-frequency dataset to compare the natural mechanisms with numerical interaction using individual conditional expectation plots. Their study found that the coupling between K-means and RF models could help in forecasting the development of the bloom-forming.

Parallelly studies started on practical information about effective monitoring systems for coastal algal blooms like Kwon et al. [10]. They used two machine learning models like

Artificial neural networks and SVM techniques to develop an optimal chlorophyll-a model.

In our study, data has been acquired from the National Oceanic and Atmospheric Administration (NOAA) http://www.class.noaa.gov/ of Visible Infrared Imaging Radiometer Suite (VIIRS) images to study factors that cause HABs. Satellite images of chlorophyll intensity serve as a useful proxy for HABs‘[12]. Furthermore, environmental data from a numerical model such as surface water temperature, salinity, and wind are considered. By themselves, these factors cannot predict either the location or the intensity of resulting blooms. We have used surface sea temperature, which is useful in providing additional contextual information for the model. The objective of this work is to provide a forecast of the HABs' manifestations in the Gulf of Mexico with a reasonable degree of confidence.

We have achieved forecasting of HABs by using a deep neural network trained on the 2-dimensional time series proxy data, chlorophyll with environmental data providing additional information for more accuracy. Moreover, the network also fills in the gaps of the time series of satellite images data along the way.

Chapter 3

PROBLEM DESCRIPTION

Satellite observations have produced an abundance of information on the ocean circulation. Figure 3.1 shows the satellite image of the area of our study. For this region we have got 200 X 200 pixel information for every hour of each day's data. In the studies related to oceanography similar to satellite observations, Ocean General Circulation Models are also useful. But regional high-resolution models are less readily available due to their computational cost, and using them for data analysis is challenging. However, ML has proved its capabilities with an efficient approach to extract the necessary knowledge from these large oceanographic datasets and to discover the hidden patterns and trends.

Also, due to the complex nature of the data acquisition from satellite images, several factors such as cloud cover over the region of interest, prevent comprehensive data acquisition, which results in missing data. The cloud cover sometimes leads to data gaps, it may be for a couple of days or the whole month. There may be partial coverage for the days when data is present. ML algorithms trained only on the available regions (limited data) would have to adapt to new regions with different physics. For addressing this sort of situation the deep neural networks are known for their ability to generalize.

Figure 3.2 shows the areas of missing data (NaNs) for a few days in June 2016 data. The image is separated with respect to the data availability. The white color represents the NaNs in the data and grey color represents the intensity of chlorophyll present in that region.

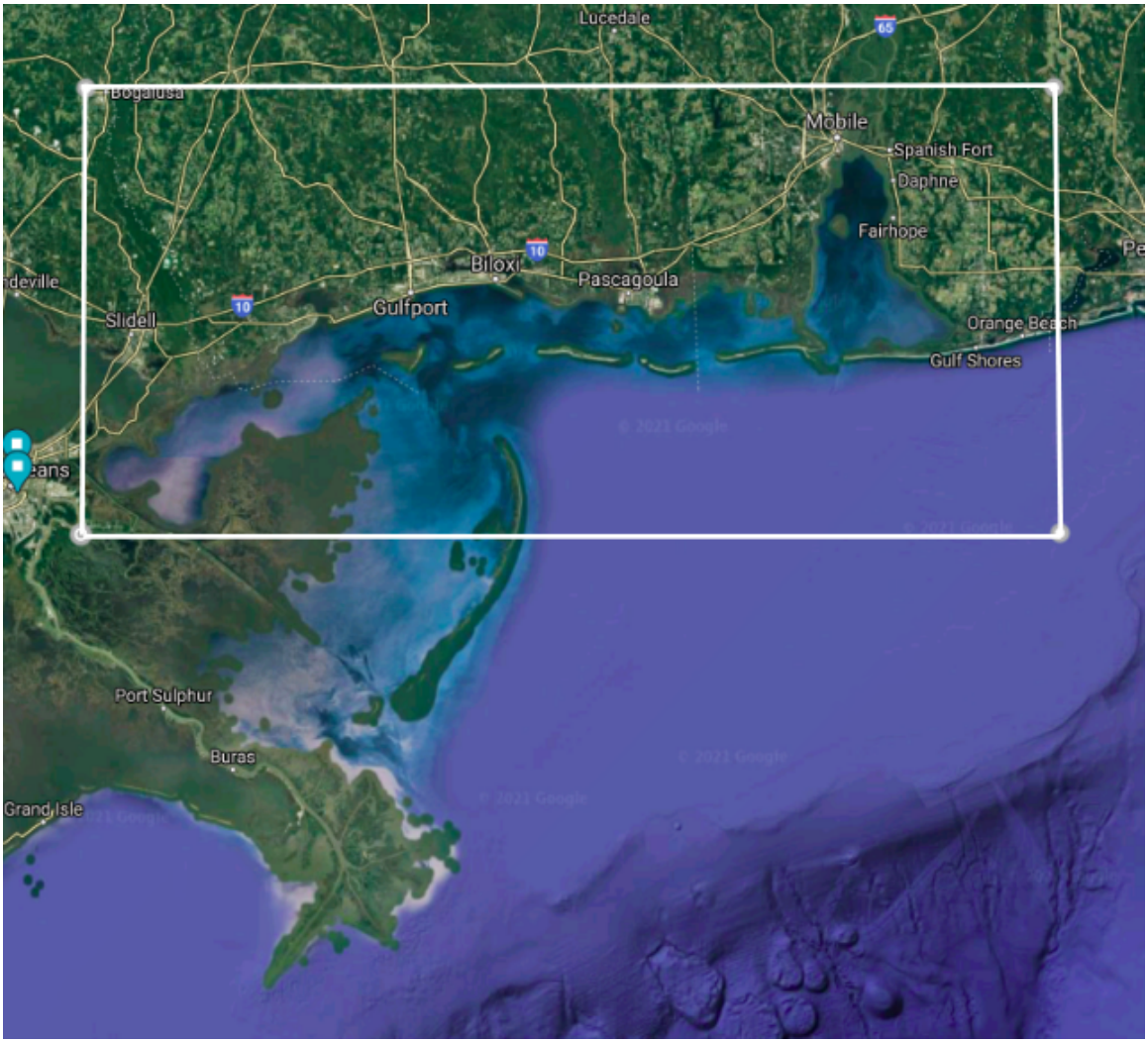

Figure 3.1: Satellite image of the area of our study is represented inside white box. Map is created using google.com/maps.

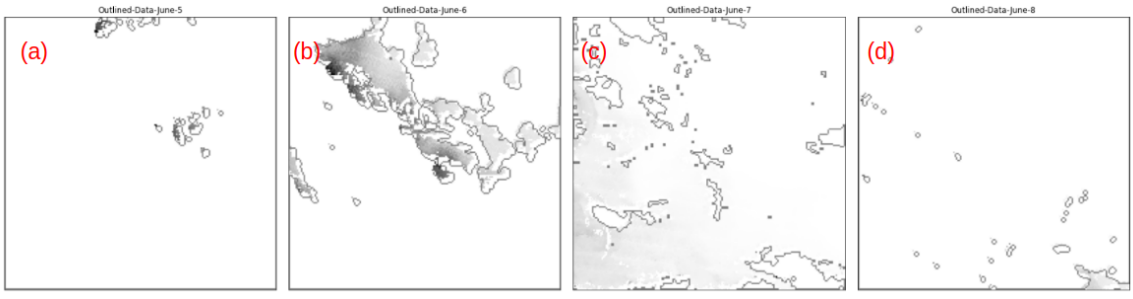

Figure 3.2: The areas of missing data (NaNs) are shown, separated with respect to the data availability for four days of June, 2016, where white color represents NaNs and intensity of color (grey) represents the intensity of chlorophyll. (a) June 5, 2016. (b) June 6, 2016. (c) June 7, 2016. (d) June 8, 2016.

Chapter 4

METHODOLOGY

The main objectives of this thesis are twofold: (a) train a prediction model to fit a time-series of HAB data presented in the form of spatial images, in order to predict the next point in the time-series; and (b) fill the missing data within the time-series used to train the model, in a way that utilizes deep insights from the data, as opposed to a trivial alternative, e.g., using interpolation.

A unique contribution of this thesis is an iterative, bidirectional training scheme, where we train *two* models, one for forward prediction, and another for backward prediction. The logic behind this choice is that if there is a functional dependence within the data in the forward time direction, then such a dependence (perhaps a different function) may also exist in the backward time direction, which may be leveraged for the purpose of predictions to fill the gaps in the data. We train each model to predict the next data point (chlorophyll concentration) in their respective time-direction, based on an LSTM recurrence over the previous L data points. Since there are missing cells in the grid within each data point, we use a custom loss function that ignores (or underemphasizes) prediction errors on missing cells. Thus the loss function critiques the models based mostly on known cells alone, while the models act with (forward/backward) predictions that are spatio-temporally consistent across both missing and visible cells, thus updating the input training data, and consequently changing the object of critique. This actor-critic training scheme progresses iteratively, leading to the iterative improvement of the models/actors, while at the same time filling the missing data with iteratively improving estimates. Our models are trained on a month-long time series, where each data point corresponds to a day, with $L = 5$.

4.1 Pre-processing

We consider two different types of data, viz., chlorophyll concentration and sea surface temperature, from the meteorological data, and process them before using them as image channels to train our models. In this section, we discuss these pre-processing steps.

Chlorophyll Data: The raw chlorophyll data comes as an hourly file of concentrations over a 2-D grid (image), and we convert it to daily resolution by averaging it.

Sea Surface Temperature Data(USM Concorde Data): The temperature data also comes as hourly resolution, and we average it to convert to the daily resolution files. Since

the 2-D grid resolution of temperature data is different from the chlorophyll data, we do bi-linear interpolation on the temperature data in order to match it with the spatial resolution of the chlorophyll data.

Next we integrate the above two images into the channel axis, and send them as input to the model. Channels are normally used to represent different colors for visual images, but we use them to hold different kinds of HAB data. From these 2-channel images, a time series generator creates inputs and outputs series for training, with look back $L = 5$. In more detail, if I_t is the 2-channel image for day t, then the time series generator creates time series X and Y, where $X_t = (I_{t-L}, I_{t-L+1}, \ldots, I_{t-1})$, and $Y_t = I_t$. We train a model to predict Y_t, given X_t as input. Using Y, we also create a weight matrix where for the regions of NaNs in Y_t we put a lower weight w_{lo}, but for other regions we put a higher weight, w_{hi}. These weights are used in weighing the model's loss function during the model update stage. The idea is that prediction *discrepancies* in unavailable (NaN) regions should not to be penalized harshly, while prediction *errors* in available regions should be.

A conceptual diagram presenting our methodology to forecast and fill missing data in the HABs is shown in Figure 4.1 and the percentages of NaNs present in june 2016 data is shown in Figure 4.2.

4.2 Training Algorithm

Our algorithm is shown in Algorithm 1.

Line 1: Initialize a forward Neural Network μ_f and a backward Neural Network μ_b with random weights. μ_f and μ_b both take a sequence of L images as input and predict the next image in the sequence.

Line 2: Repeat the section from Line 3 to Line 13.

Line 3, Line 4, and Line 5: Let T be the number of images in the input data. We train the forward model, μ_f with a sequence of L images starting from I_1. That means, $(I_1, I_2, \ldots I_L)$ is the first input with (I_{L+1}) as the first output. $(I_2, \ldots I_{L+1})$ is the second input with (I_{L+2}) as the second output. $(I_{T-L}, \ldots I_{T-1})$ is the last input with (I_T) as the last output.

Line 6, Line 7, Line 8: We train the backward model μ_b with the image sequence reversed. Here, $(I_T, I_{T-1}, \ldots I_{T-L+1})$ is the first input (I_{T-L}) is the first output. Similarly, $(I_{L+1}, I_L, \ldots I_2)$ is the last input and (I_1) is the last output.

Line 9, Line 10, Line 11: Next we predict the values of the missing pixels using both the forward and backward models.

1. For each image in $(I_1, I_2, \ldots I_L)$ with missing pixels, the NaNs are replaced by a convex combination (with α) of the previous iteration's predictions with the current

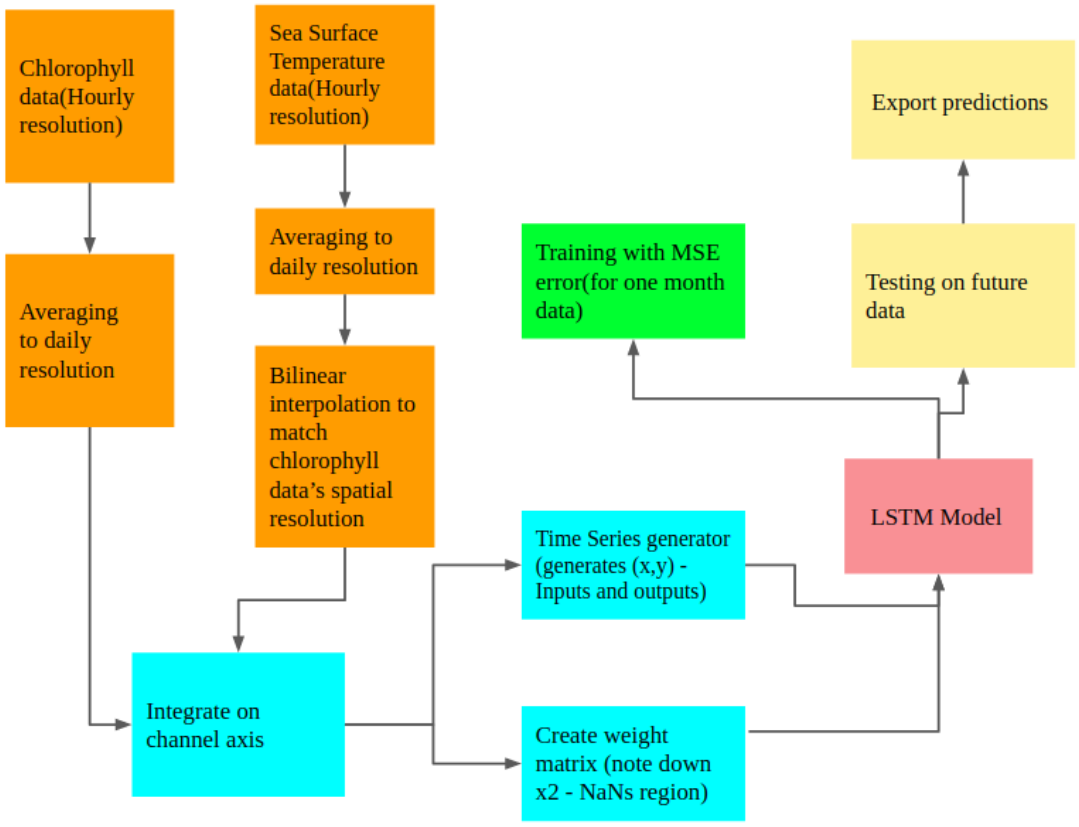

Figure 4.1: Conceptual diagram presenting the methodology used to forecast the Harmful Algal Blooms (HABs). The boxes in orange indicates pre-processing the data, The boxes in blue indicates the data generation flow to the red box which indicates the LSTM model followed by test and training results indicated in yellow and green boxes respectively.

predictions of the backward model, because the forward model only predicts from I_{L+1} to I_T.

2. For each image in $(I_{T-L+1}, \ldots I_T)$ with missing pixels, the NaNs are replaced by a convex combination (with α) of the previous iteration's predictions with the current predictions of the forward model, because the backward model only predicts from I_{T-L} to I_1.

3. For each image in $(I_{L+1}, I_{L+2}, \ldots I_{T-L})$ with missing pixels, the NaNs are replaced by a convex combination of the previous iteration's predictions with the average of the predictions from the forward and the backward model.

Line 12: For the forward model, μ_f, back-propagate the gradients generated using the custom loss function

MSE = $\sum_{j=l+1}^{T} [\sum_{h \notin M_j} w_{hi}(F_j[h] - I_j[h])^2 + \sum_{h \in M_j} w_{lo}(F_j[h] - I_j[h])^2] / (T - l)$

The loss is computed using a formula that computes Mean Square Error (MSE) between

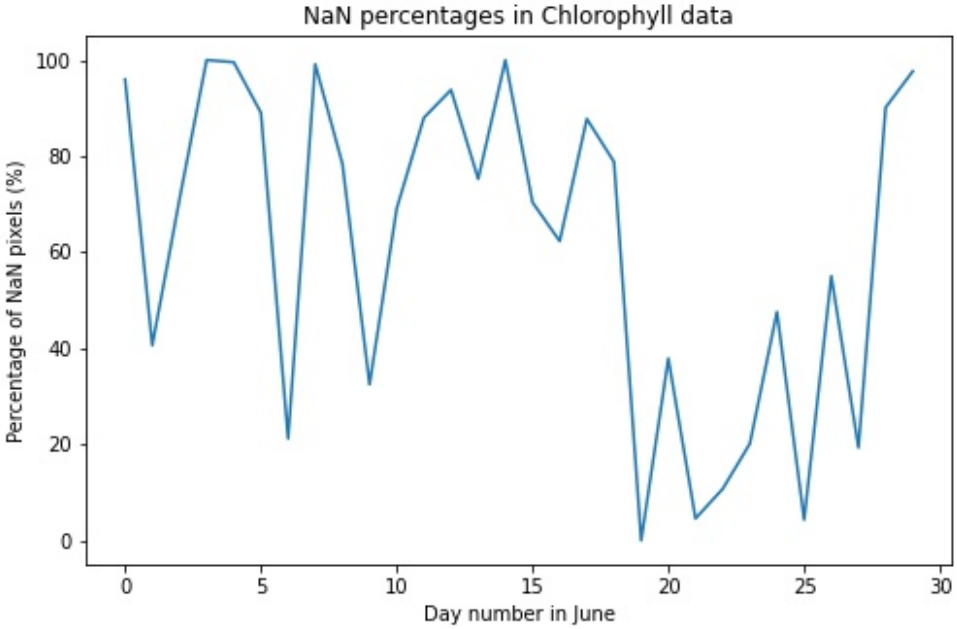

Figure 4.2: This shows the number of NaNs percentage with respect to number of days for 30 days of June data starting from June 1, 2016.

the true intensities and predictions of the available pixels and also between the previous iteration's predictions and the current iteration's predictions. The idea behind adding the second expression is to avoid large changes in the gradients.

Line 13: For the backward model, μ_b, back-propagate the gradients generated using the custom loss function

MSE= $\sum_{j=1}^{T-l}[\sum_{h\notin M_j} w_{hi}(B_j[h]-I_j[h])^2+\sum_{h\in M_j} w_{lo}(B_j[h]-I_j[h])^2]/(T-l)$

The loss is computed in an identical way as the forward model.

Line 14: Repeat the loop from Line 3 to Line 13 until the MSE is below a tolerance threshold.

Line 15: Return the forward model μ_f, to provide forecast.

4.3 Models

Multiple feed-forward neural network models were evaluated by varying the layers, activation functions, and normalization performed across the axis. Four model configurations that are evaluated in this thesis are shown in Figures 4.3, 4.4. Via experiments with these 4 models (see next chapter), we found that model D (Figure 4.4 right) achieves the best accuracy. The framework of model D is shown in Figure 4.5. Next, we describe the purpose

Algorithm 1 Bi-directional Training for Time-Series

Require: Time series of images $I_1, I_2, \ldots I_T$; the set of missing pixels in each, M_i for I_i; variable l for look back; learning rate α for missing pixels; weights for missing (w_{lo}) and non-missing (w_{hi}) pixels. Missing pixels are initialized to 0.

Ensure: Trained NN model, μ_f, for forward prediction

1: $\mu_f, \mu_b \leftarrow$ Initialize forward and backward NN models randomly. These models take a sequence of l images as input, and predict the next image in the sequence.
2: **repeat**
3: **for** $j \leftarrow l+1, l+2, \ldots, T$ **do**
4: $F_j \leftarrow \mu_f(I_{j-l}, I_{j-l+1}, \ldots, I_{j-1})$
5: **end for**
6: **for** $j \leftarrow T-l, T-l-1, \ldots, 1$ **do**
7: $B_j \leftarrow \mu_b(I_{j+l}, I_{j+l-1}, \ldots, I_{j+1})$
8: **end for**
9: **for** $j \leftarrow 1, 2, \ldots, T$ **do**
10: $$\forall h \in M_j, I_j[h] \leftarrow (1-\alpha) I_j[h] + \alpha \cdot \begin{cases} B_j[h] & \text{if } 1 \le j \le l \\ \frac{1}{2}\left(B_j[h] + F_j[h]\right) & \text{if } l+1 \le j \le T-l \\ F_j[h] & \text{if } T-l+1 \le j \le T \end{cases}$$
11: **end for**
12: Update μ_f with MSE=$\sum_{j=l+1}^{T} [\sum_{h \notin M_j} w_{hi}(F_j[h] - I_j[h])^2 + \sum_{h \in M_j} w_{lo}(F_j[h] - I_j[h])^2]/(T-l)$.
13: Update μ_b with MSE=$\sum_{j=1}^{T-l} [\sum_{h \notin M_j} w_{hi}(B_j[h] - I_j[h])^2 + \sum_{h \in M_j} w_{lo}(B_j[h] - I_j[h])^2]/(T-l)$.
14: **until** Total MSE is below a tolerance threshold
15: **return** Trained forward model, μ_f

of each layer used in model D.

Convolutional LSTM2D: Since our predictions are based on time series data we use Long short-term memory (LSTM) network [8]. LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. Convolutional LSTM2D is similar to an LSTM layer, but the input transformations and recurrent transformations are both convolution. In our best model(Model D) we use ConvLSTM2D layer with filters=20, kernel-size=(5,5) and rectified linear activation function or ReLU activation function.

Convolutional 2D: A convolution layer creates a kernel (filter) that convolves with the layer input to produce a tensor of outputs. It is used to extract features from the image or parts of an image.

Model A:	**Model B:**
x = ConvLSTM2D(inputs) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) out = Dense(x) out = multiply([out,const])	x = ConvLSTM2D(inputs) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = BatchNormalization(axis=[0,1,2])(x) x = MaxPooling2D(x) x = Conv2D(x) x = BatchNormalization(axis=[0,1,2])(x) out = Dense(x) out = multiply([out,const])

Figure 4.3: Model configuration, where BatchNormalization layers is added to Model A to create Model B

Convolutional 2D Transpose: Transposed convolutions or deconvolution layer is used to up sample the feature maps to preserve the input shape (since the output shape is identical to input shape and conv2d layer reduces the feature map size). Transposed convolutional layer also preserves the connectivity of the feature maps.

Max Pooling 2D: Maxpooling layers sub-samples the input feature map and focuses on the most important features in the data (taking the max operation)

Batch Normalization: BN layers normalizes the inputs of the layer over the axes of the feature maps (specified in the arguments). This enables the network to consider all features with equal importance. This would stabilize the learning process.

Dense: Dense layer or fully connected layer that takes a flattened form of the features and interconnects the nodes (with weights). This will enable the information within the feature map stand out.

Multiply: A multiplication operation (with a constant) to the layer inputs. The purpose is to scale the output values to a constant range reflecting the range of pixel values, viz., 0-255.

Model C:	**Model D:**
x = ConvLSTM2D(inputs) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = Conv2D(x) x = BatchNormalization(axis=[0,1,2])(x) x = MaxPooling2D(x) x = Conv2DTranspose(x) x = Conv2D(x) x = BatchNormalization(axis=[0,1,2])(x) out = Dense(x) out = multiply([out,const])	x = ConvLSTM2D(inputs) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2D(x) x = MaxPooling2D(x) x = Conv2DTranspose(x) x = BatchNormalization(axis=[1,2])(x) out = Dense(x) out = multiply([out,const])

Figure 4.4: Model configuration C, where Conv2DTranspose layer is added to model B from Figure 4.3 and the model configuration D, similar as model configuration B in Figure 4.3, but in model B the Batch Normalization is done across axis=[0,1,2] where as in model D the BatchNormalization is performed across only axis=[1,2]. Also, in model D unlike other models only two max pooling layers, two convolution 2D layers are used.

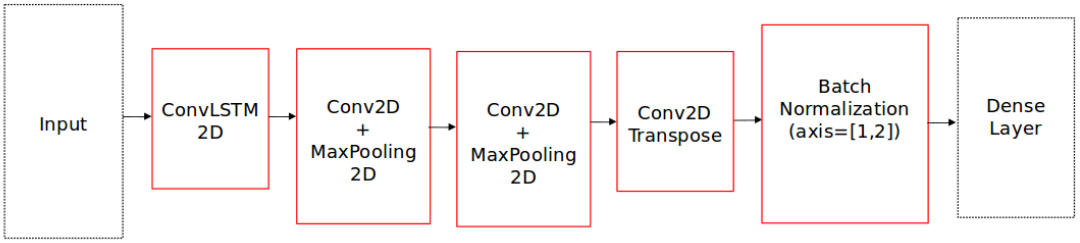

Figure 4.5: Model Framework.

Chapter 5

EXPERIMENTAL RESULTS

5.1 Model Accuracy

Time series data requires preparation before training a deep learning model. A supervised learning algorithm requires that data is provided as a collection of samples, where each sample has an input component (X) and an output component (Y). A time series must be transformed into samples with input and output components. The transform both informs what the model will learn and how we intend to use the model in the future when making predictions, e.g. what is required to make a prediction (X) and what prediction is made (Y).

The models A-D (Figures 4.3, 4.4 in Chapter 4) were trained with the following parameters:

Lookback : Lookback is the number of previous days observations to use in the input portion of each sample, In our models we consider lookback as 5. This is variable l in Algorithm 1.

Iterations: The model is trained for a fixed number of iterations. We perform 1000 iterations on our model. This is in place of repeat-until in steps 2 to 11 in Algorithm 1.

Epoch : An epoch is an iteration over the entire data for updating a model. We use 4 epochs to train our model. That is, steps 12 and 13 of Algorithm 1 are repeated 4 times within each outer iteration.

Optimizer : Optimizer helps to reduce the loss values by changing the attributes of the neural network such as weights and learning rate. It also helps to get results faster. We used Adam optimizer in our training.

w_{hi} **:** Weightage in the loss function for MSE in predicting non-missing pixels. For our model the w_{hi} is 0.95.

w_{lo} **:** Weightage in the loss function for MSE in predicting the missing pixels w.r.t. the previous iteration's prediction. For our model the w_{lo} is 0.05.

Alpha(α) : update rate of the present iteration's prediction of the missing pixels. The α value is 0.05 in our model.

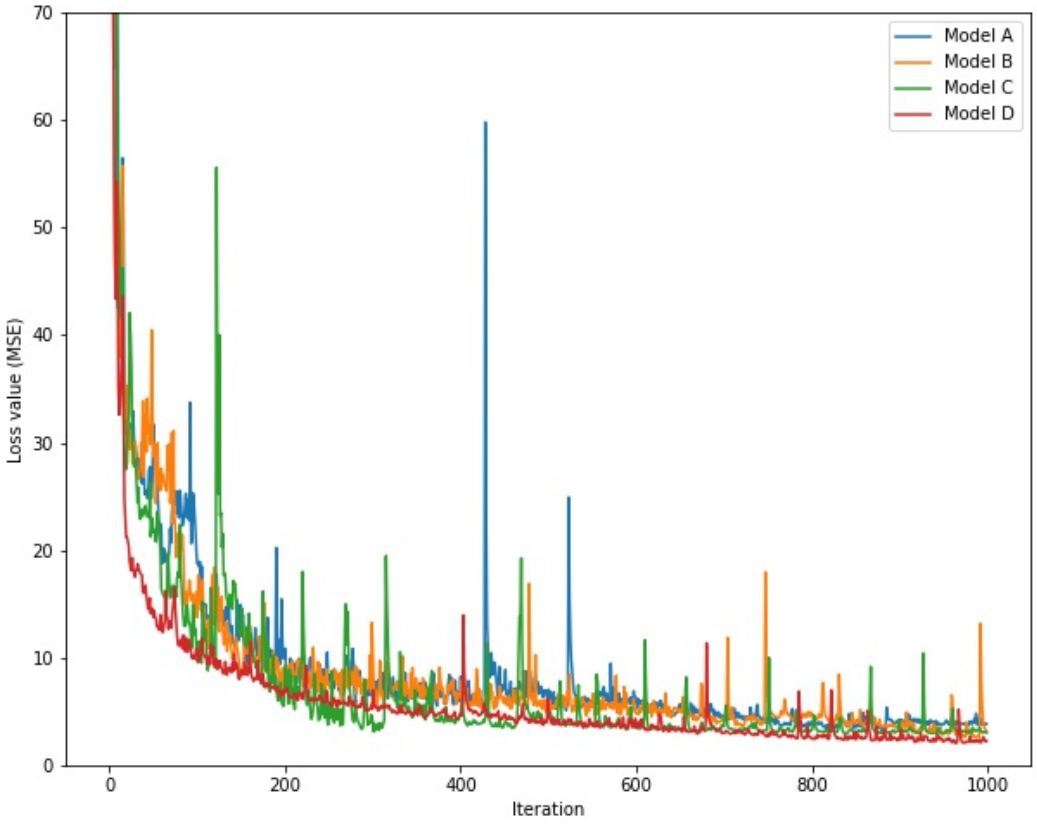

Figure 5.1: The Loss Values were calculated using Mean Square Error (MSE) for the respective iteration during training of models.

Figure 5.1 shows the loss values obtained over 1000 iterations of training of models A-D. Since model D shows lower loss values, we use this model for the remaining analyses. Its accuracy measured as root-mean-square deviation between predicted and true values, with a total pixel value range of $[0, 255]$ is 99.27%.

5.2 Filling Missing Pixels

Step 10 in Algorithm 1 fills missing pixels using both forward and backward models, iteratively. We save these completed images for use in forward prediction during the test phase. In this section, we investigate the algorithm's ability to fill missing pixels. After the 1000th iteration, we take the forward model and predict the full set of pixels for days June 26-28. Since each prediction requires a look-back of 5 days in input, we use the saved

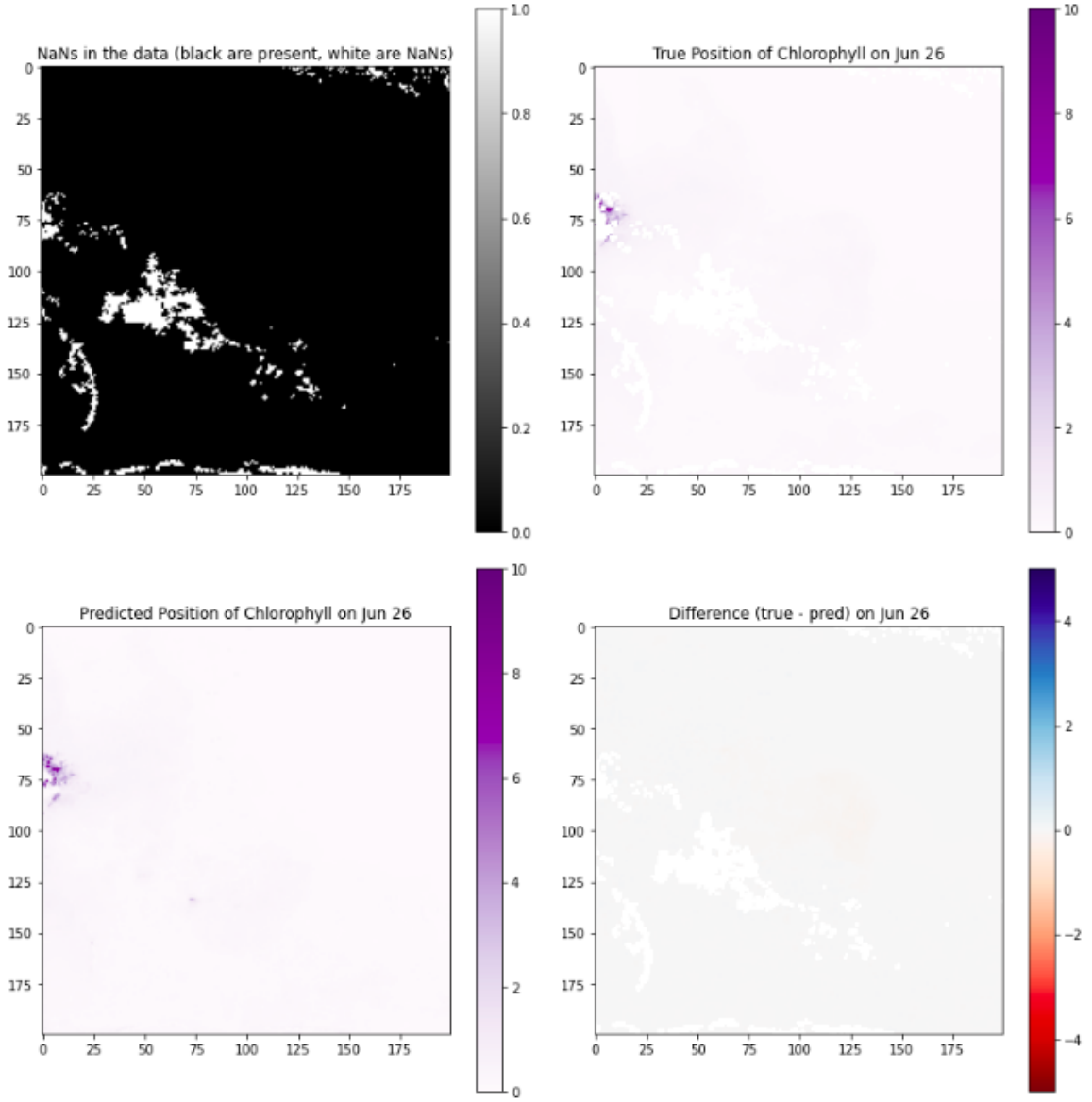

Figure 5.2: The original, True, Prediction and Difference of True-Prediction Results of June 26, 2016.

complete images for these inputs. This allows the prediction to be based on complete inputs, rather than the original data that contains missing pixels.

Figures 5.2, 5.3, 5.4 show the results for our chosen days. Each consists of four sub-images, described below.

Top-Left Sub-Image: Image representing the NaN's in the data, where black represents the regions where chlorophyll information is present and white represents NaNs (missing pixels).

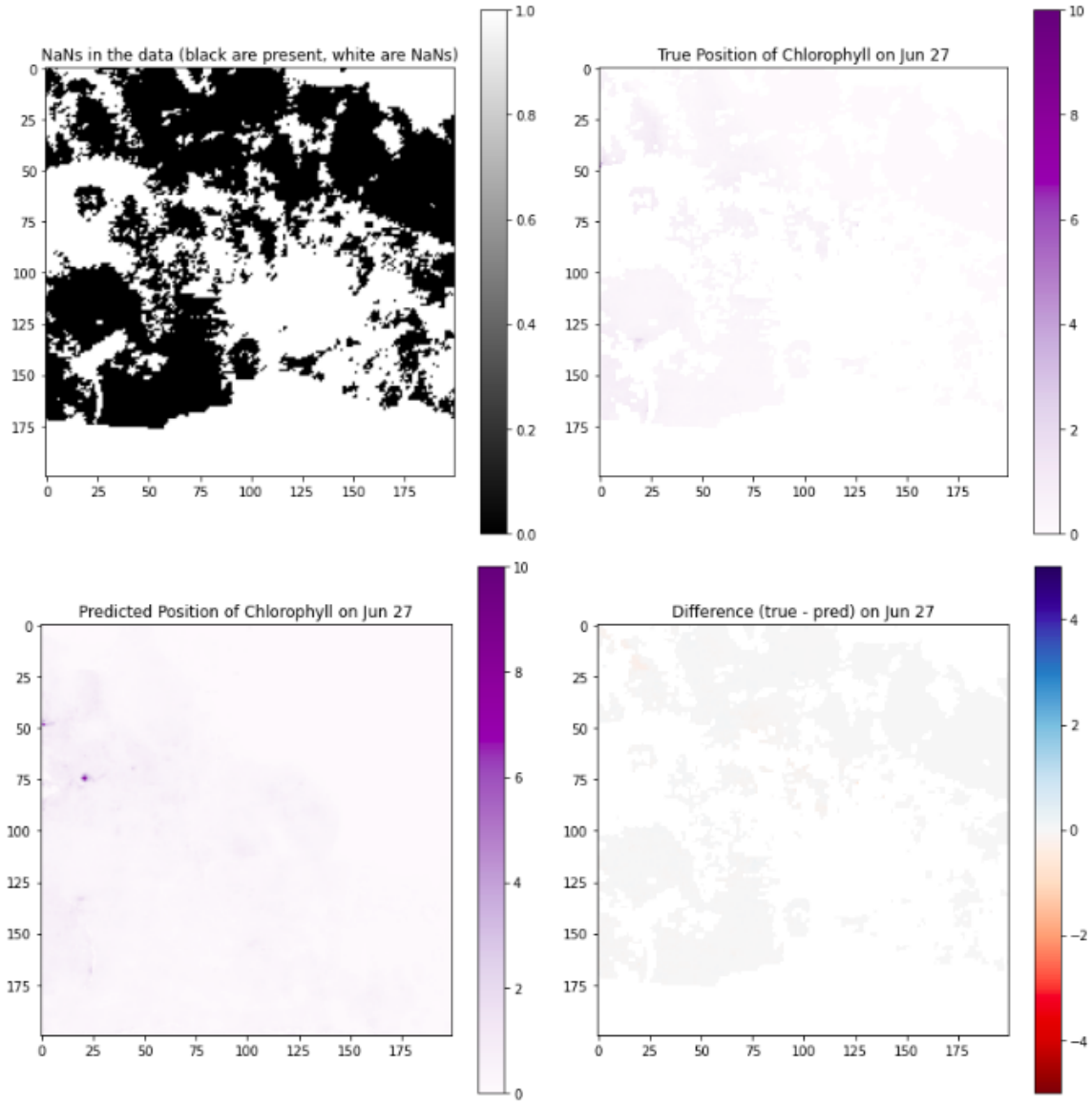

Figure 5.3: The original, True, Prediction and Difference of True-Prediction Results of June 27, 2016.

Top-Right Sub-Image: This image shows the true position of chlorophyll, where the intensity of the color represents the intensity of chlorophyll present in that region.

Bottom-Left Sub-Image: This is the predicted image of chlorophyll for the specific day using the trained forward model. In this sub-image of Figures 5.2, 5.3, 5.4, we see that the model's predictions match well the true intensities, where available. *More interestingly, the model predicts high intensities in certain regions where no original data is available. This is especially pronounced in Figure 5.4. The model thus reveals previously unknown algal activity.*

Bottom-Right Sub-Image: This image represents the difference between the true image and the predicted image. For this difference we have masked the regions of the predicted image where there are NaN's initially to compare. Almost negligible difference in Figures 5.2, 5.3, 5.4 means that the model's accuracy of predicting the algal activity is high.

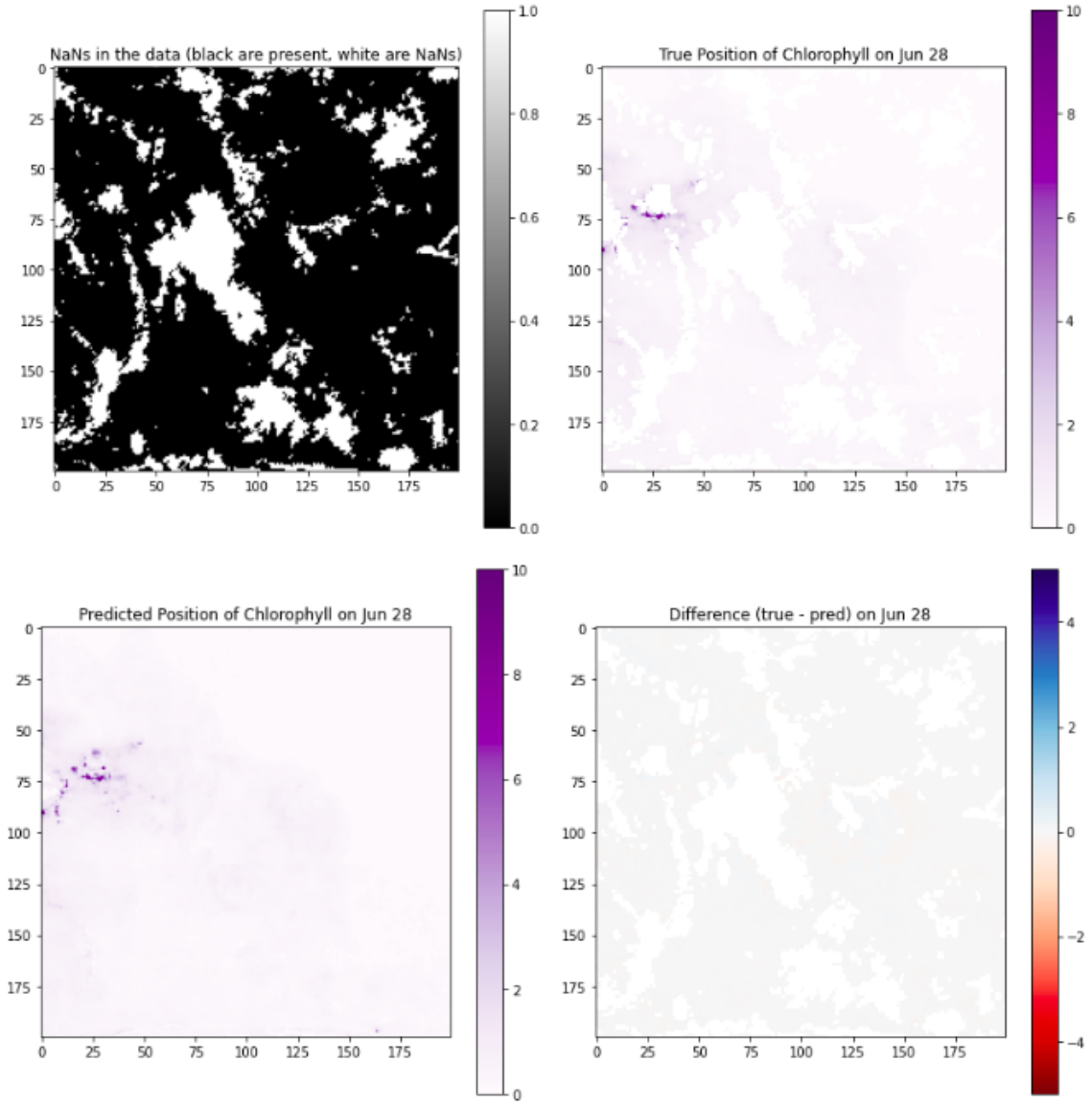

Figure 5.4: The original, True, Prediction and Difference of True-Prediction Results of June 28, 2016.

5.3 Prediction of Next in Sequence

Here we study the model's ability to *forecast*, i.e., predict the next day in sequence outside of the training sequence. In order to show the forecast results we have trained the model by varying the number of training days, viz., with 10, 15, 20, and 25 days. Using the resulting models, we respectively forecast days 11, 16, 21, and 26. Since the true data is also known for these days, we then compare the forecast with the true data.

For prediction results on June 11,2016: We have trained the model with 10 days of June, 2016 data starting from June 1, 2016. Then we used this training results to predict the June 11, 2016 data. Both the true position of June 11, 2016 and predicted image of June 11, 2016 are shown in Figure 5.5. With very few training days (viz., 10) the model can accurately predict the intensity of chlorophyll concentration in some areas. However the model is unable to predict all of the regions where the algal activity is present in the true data.

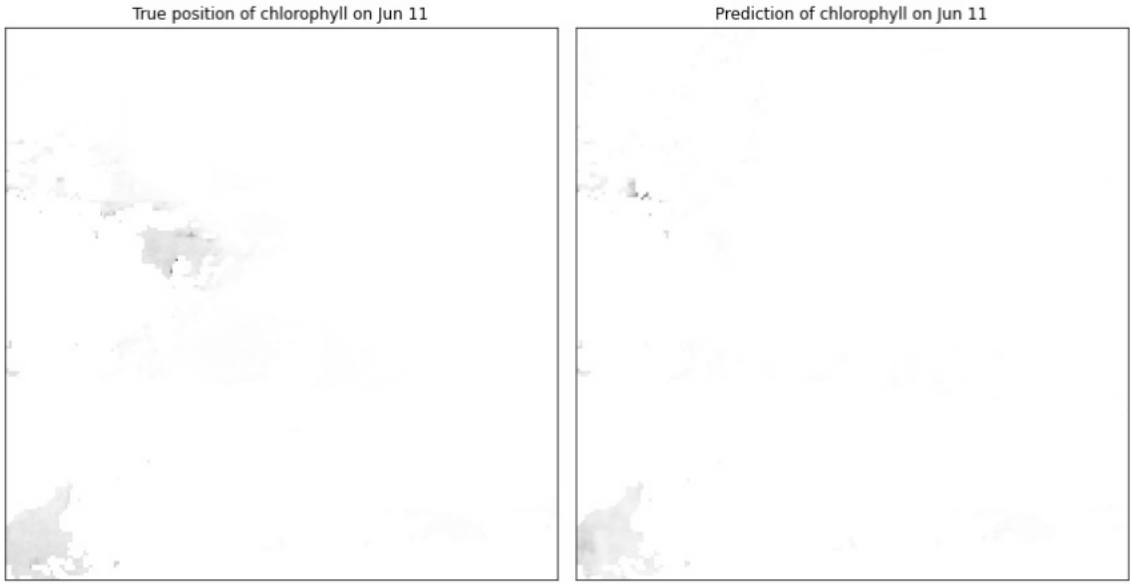

Figure 5.5: True and the predicted images of June 11,2016

For prediction results on June 16, 2016: We have trained the model with 15 days of June, 2016 data starting from June 1, 2016. Then we used this training results to predict the June 16, 2016 data. Both the true position of June 16, 2016 and predicted image of June 16, 2016 are shown in Figure 5.6. The model is able to predict most of the regions where the algal activity is present. However the model is unable to predict the *intensity* of chlorophyll concentration in all of the regions. The forecast intensities

are significantly lower than the true values, which we believe indicates an unexpected deviation in the training data from whatever pattern the model has gleaned.

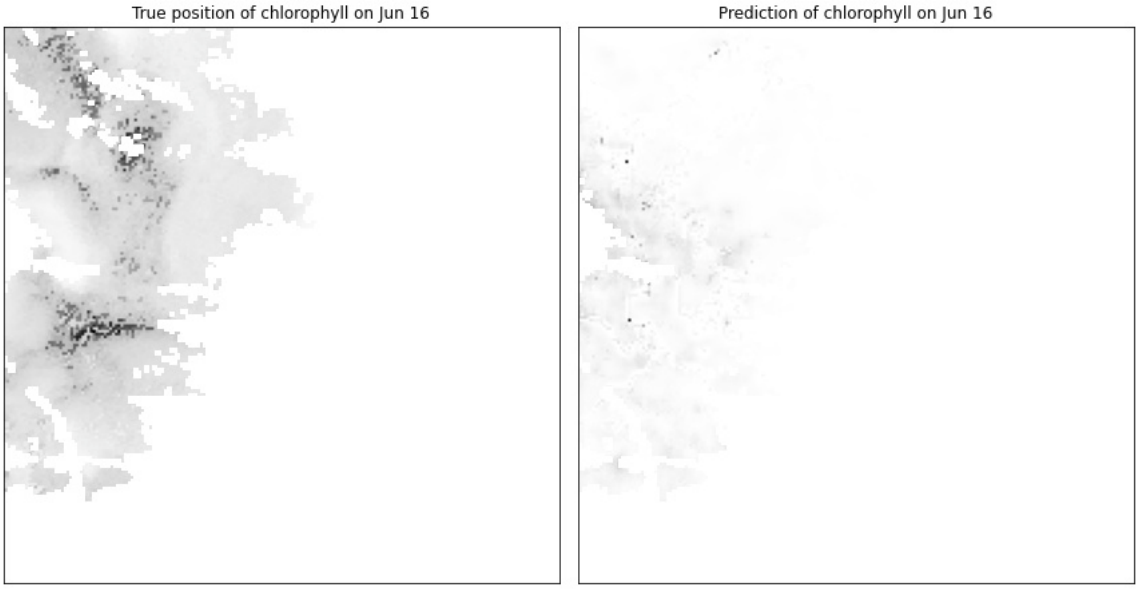

Figure 5.6: True and the predicted images of June 16,2016

For prediction results on June 21, 2016: We have trained the model with 20 days of June, 2016 data starting from June 1,2016. Then we used this training results to predict the June 21, 2016 data. Both the true position of June 21, 2016 and predicted image of June 21, 2016 are shown in Figure 5.7. Here again, we see that the forecast intensities are significantly lower than the true values, continuing a trend of pattern breakage that we observed in Figure 5.6. We believe there are *hidden* factors that are affecting the algal activity, extraneous to the training data, which is why the model is unable to account for them.

For prediction results on June 26, 2016: We have trained the model with 25 days of June, 2016 data starting from June 1, 2016. Then we used this training results to predict the June 26,2016 data. Both the true position of June 26, 2016 and predicted image of June 26, 2016 are shown in Figure 5.8. The model prediction when trained over 25 days appears better than 15 or 20 days. The model is able to predict almost all of the regions where the chlorophyll is present, but fails to predict the intensity of chlorophyll concentration, similarly to Figures 5.6 and 5.7. Since all of the days from June 01, 2016 till June 25, 2016 days data is also a part of this training data, our hypothesized pattern breakage due to hidden factors between June 16 and 20, 2016

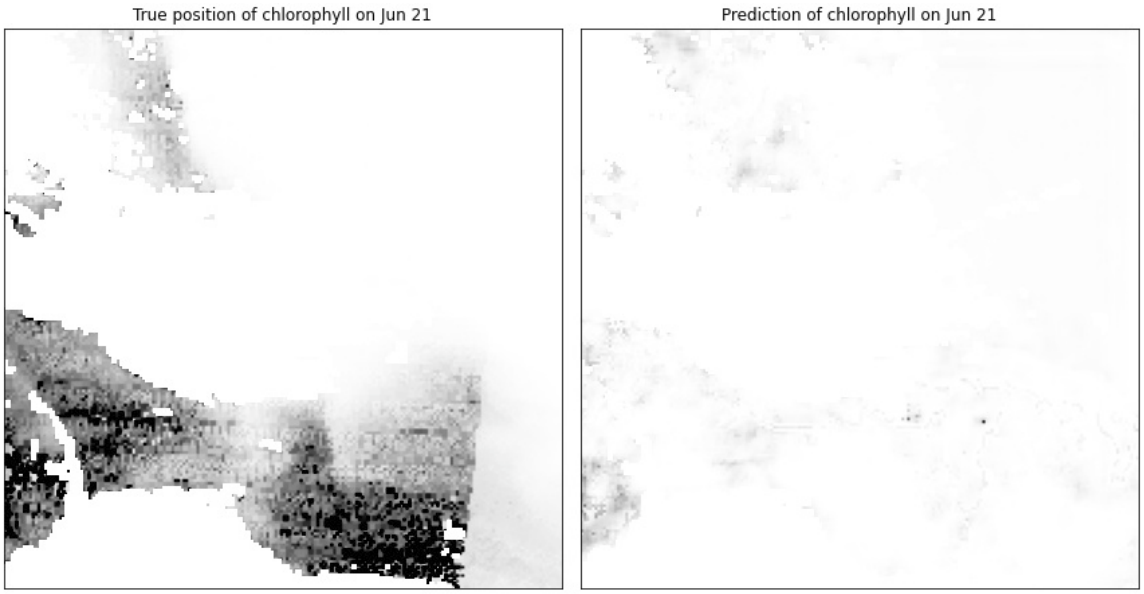

Figure 5.7: True and the predicted images of June 21,2016

data would also have impacted this forecast. However, the difference in intensities is not as large as in Figures 5.6 and 5.7, which might indicate that the influence of our hypothesized hidden factors may have waned to some extent by June 25.

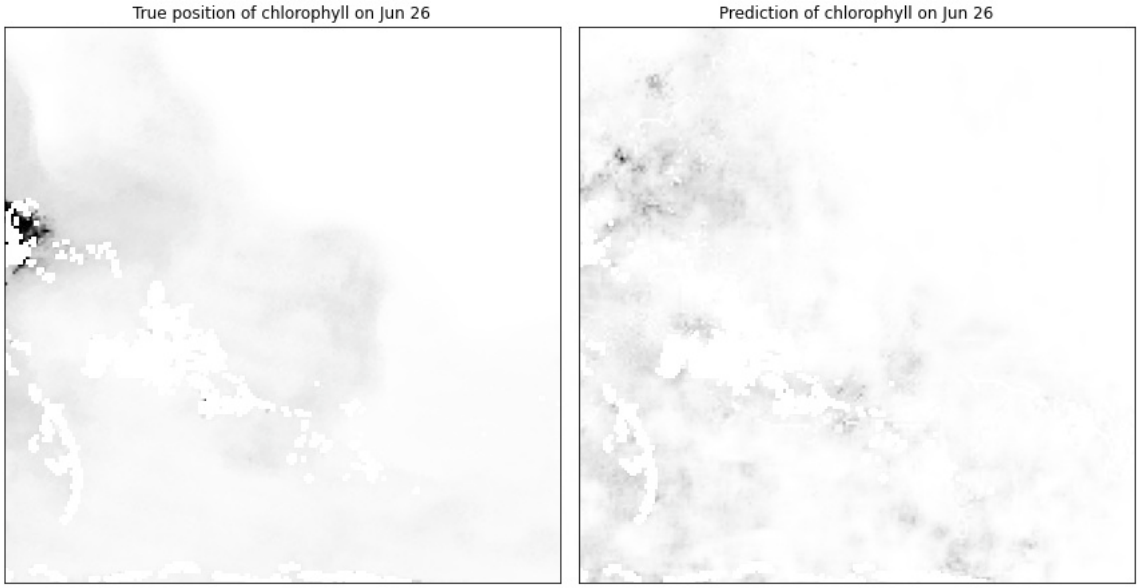

Figure 5.8: True and the predicted images of June 26,2016

5.4 Sensitivity to the Amount of Training Data

Having trained 4 forward prediction models, μ_f^T with $T = 10$, 15, 20, and 25 days of time-series data respectively, for the previous experiment in Section 5.3, we now have the opportunity to investigate the influence of varying amounts of training data on the accuracy of predictions. In this section, we measure prediction error of model μ_f^T over days $6, \ldots, 30$ of June 2016, for only the non-missing pixels, using the following formula:

$$E(T) = \frac{1}{(30-6+1)} \sum_{j=6}^{30} \frac{1}{|I_j| - |M_j|} \sum_{h \notin M_j} \left| I_j[h] - \mu_f^T(I_{j-5}, \ldots, I_{j-1})[h] \right|, T = 10, 15, 20, 25,$$

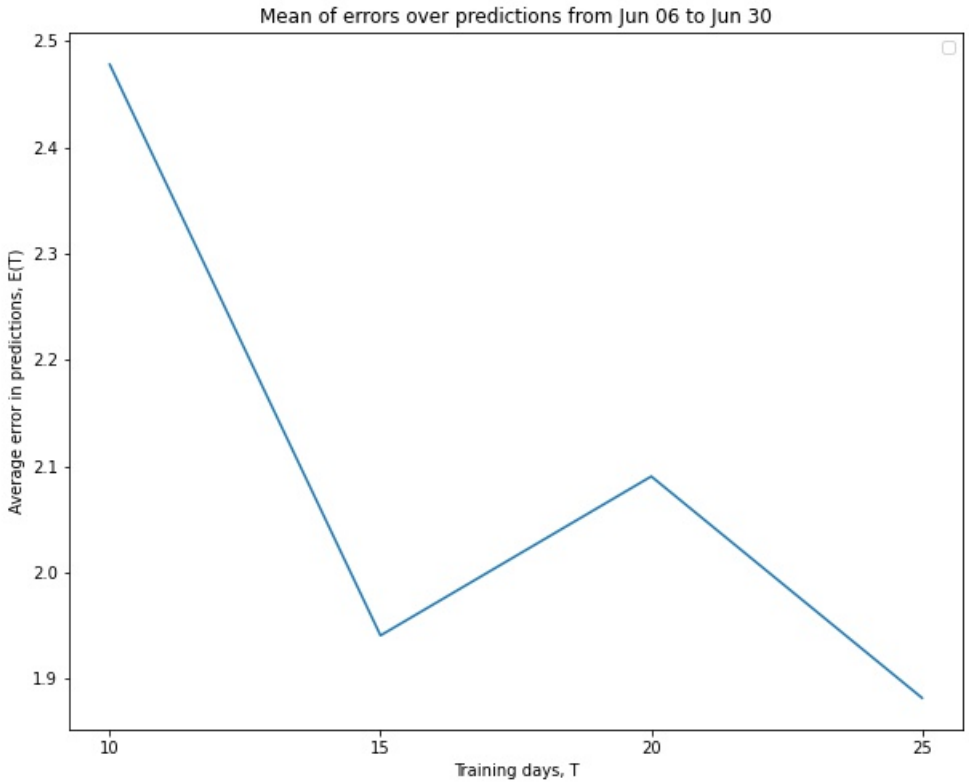

Figure 5.9: Mean prediction error E(T) with respect to training days (T).

where j represents the day, I_j is the jth day's image (data), M_j is the set of missing pixels in I_j such that h ranges over only the non-missing pixels, and the forward model μ_f^T is executed with the same look-back of $L = 5$ as during training. Thus $E(T)$ measures the average prediction error per visible pixel, per day of prediction. Figure 5.9 shows the plot of $E(T)$ vs. T, for $T = 10$, 15, 20, and 25.

One might expect that with increasing amounts of training data, i.e., with increasing T, the prediction error $E(T)$ would decrease monotonically. Although the trend in Figure 5.9 is largely consistent with that expectation, we nevertheless see a deviation around days 15-20. We believe this deviation *confirms our hypothesis in Section 5.3 that our training data is missing important (hidden) factors that strongly influence the algal activity during precisely this period, June 15-20.* Once this period of extraneous activity is over, we see a return to the expected trend in Figure 5.9 for $T = 25$, where the lowest prediction error is recorded.

Chapter 6

CONCLUSIONS

Our experimental studies have shown that a Machine Learning model using Long Short Time Memory layer can perform well in terms of filling missing data in HAB timeseries data. The model not only achieves high accuracy in predicting the available data, but also high spatial consistency when predicting the missing data. Furthermore, the former allows us to have a high confidence in the efficacy of the latter. Hence, we posit our model and training approach as an effective means to overcome practical limitations such as cloud cover in geosciences and remote sensing.

When it comes to forecast of HAB intensity outside of the training range, however, our model does not perform quite well. This is especially the case somewhere in the middle of the timeseries. The experiment on the sensitivity to the amount of training data further reinforces this point. It appears clear that there are extraneous factors influencing the algal activity, that are not present in the training data. Therefore, further studies may need to consider additional factors apart from temperature as supporting data for chlorophyll, e.g., wind curl, back scattering, etc. It would be rather simple to integrate such additional data as additional channels in the input, and this may improve the model's performance with foreceast. Another potential avenue is to use modeling techniques that explicitly account for hidden factors, such as hidden Markov models [4].

Finally, from the experiments on the sensitivity to the amount of training data, it is clear that the more the amount of training data used, the better the prediction results. This is consistent with the common wisdom of "data hungry" models in machine learning. However, longer timeseries also require longer to train. Our experiments were run on state-of-the-art GPUs (Nvidia Quadro RTX 6000) over multi-day training phases, and this would prolong further if, say, we were to train over a year's worth of data rather than a month. Unfortunately, such computational resources are also exceptionally power hungry. Therefore, our need for better accuracy and performance should be balanced against ethical considerations, specifically the environmental impact of computing.

BIBLIOGRAPHY

[1] Sediment plume in lake pontchartrain, 2011. https://earthobservatory.nasa.gov/images/50674/.

[2] NCCOS Assists Response to Cyanobacterial Blooms in Lake Pontchartrain Caused by Opening Bonnet Carre Spillway, Jul 2020. https://coastalscience.noaa.gov/news/nccos-assists-response-to-cyanobactrial-blooms-in-lake-pontchartrain-caused-by-opening-the-bonnet-carre-spillway/.

[3] Hafez Ahmad. Machine learning applications in oceanography. *Aquatic Research, 2(3)*, pages 161–169., 2019.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

[5] Katherine Derner, Karen Kavanaugh, and Michelle Tomlinson. Refining Karenia brevis Detection: An Assessment of Ensemble Imagery Products for Operational Forecasting. In *Presented at the 8th Symposium on Harmful Algae in the U.S*. HAB-OFS publications, November 2015.

[6] Jonathan Derot, Hiroshi Yajima, and Stéphan Jacquet. Advances in forecasting harmful algal blooms using machine learning models: A case study with planktothrix rubescens in lake geneva. *Harmful Algae*, 99:101906, 2020.

[7] Paul R. Hill, Anurag Kumar, Marouane Temimi, and David R. Bull. HABNet: Machine Learning, Remote Sensing-Based Detection of Harmful Algal Blooms. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3229–3239, 2020.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[9] Analise Keeney, Karen Kavanaugh, and Katherine Derner Ed Davis. Assessment of the Gulf of Mexico Harmful Algal Bloom Operational Forecast System: An Analysis of Transport and Respiratory Irritation, 2010-2015. In *Presented at the 8th Symposium on Harmful Algae in the U.S*. HAB-OFS publications, November 2015.

[10] Yong Sung Kwon, Seung Ho Baek, Young Kyun Lim, JongCheol Pyo, Mayzonee Ligaray, Yongeun Park, and Kyung Hwa Cho. Monitoring coastal chlorophyll-a concentrations in coastal areas using machine learning models. *Water*, 10(8), 2018.

[11] David J. Lary, Amir H. Alavi, Amir H. Gandomi, and Annette L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10, 2016. Special Issue: Progress of Machine Learning in Geosciences.

[12] Sangmok Lee and Donghyun Lee. Improved prediction of harmful algal blooms in four major south korea's rivers using deep learning models. *International Journal of Environmental Research and Public Health*, 15(7), 2018.

[13] Xiu Li, Jin Yu, Zhuo Jia, and Jingdong Song. Harmful algal blooms prediction with machine learning models in Tolo Harbour. *2014 International Conference on Smart Computing*, pages 245–250, 2014.

[14] Michael T. Walsh and Martine de Wit. Chapter 45 - sirenia. In R. Eric Miller and Murray E. Fowler, editors, *Fowler's Zoo and Wild Animal Medicine, Volume 8*, pages 450–457. W.B. Saunders, St. Louis, 2015.

[15] Hye-Suk Yi, Sangyoung Park, Kwang-Guk An, and Keun-Chang Kwak. Algal bloom prediction using extreme learning machine models at artificial weirs in the nakdong river, korea. *International Journal of Environmental Research and Public Health*, 15(10), 2018.