Algal Bloom Forecasting using Remote Sensing with Spatially and Temporally Sparse Satellite Data

Einar de Gruyl[1]

Supervisor(s): dr. Jan van Gemert[1]**, Attila Lengyel**[1]**, Robert-Jan Bruintjes**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 29, 2023

Name of the student: Einar de Gruyl
Final project course: CSE3000 Research Project
Thesis committee: dr. Jan van Gemert, Attila Lengyel, Robert-Jan Bruintjes, Koen Langendoen

Abstract

This research presents a method for forecasting algal blooms using remote sensing with spatially and temporally sparse satellite data. The method involves the use of multiple interpolation methods to interpolate the sparse input data. The approach is shown to be effective in predicting algal blooms in areas where data is sparse, and the results demonstrate the potential for using this method to improve the forecasting and management of harmful algal blooms.

1 Introduction

Algal blooms are a widespread environmental problem that can have significant impacts on aquatic ecosystems and potentially pose risks to human health [1]. Algal blooms occur when there is an excess of nutrients in a body of water, causing rapid growth of algae. This can lead to decreased water quality and reduced oxygen levels. In recent years, algal blooms have become more frequent and severe, making the need for effective monitoring and forecasting methods essential [1].

Traditionally, algal blooms have been measured using in-situ measurements. However, this approach has several limitations, including needing frequent and expensive field visits, the potential for human error, and the inability to monitor large areas. Alternatively, remote sensing with satellite data, from satellites such as Landsat [2] and Sentinel-2 [3], offers a solution for monitoring these algal blooms on a large scale. Satellite data provides information on the spatial and temporal distribution of algal blooms, allowing for real-time monitoring and forecasting of algal blooms.

However, there are several challenges associated with using satellite data for algal bloom monitoring. First, satellite data is often spatially and temporally sparse, meaning that there are gaps in the spatial and temporal coverage of the data. Second, algal blooms can be difficult to detect in satellite data, as they often have similar characteristics to other aquatic features.

Therefore the goal of this research is to explore and evaluate multiple methods of handling this sparsity in the dataset to predict chlorophyll-a concentrations, which is an indicator of algal blooms, in a 5-day prediction horizon.

In this research, first, the related work in the field will be discussed in section 2. Next, the methodology used for running the experiment will be described in section 3. Then the experimental setup and results will be shown in section 4. After that, the results will be discussed in section 6. Finally, the research will be concluded in section 7.

1.1 Data Acquisition

Satellite data for algal blooms were collected over a period of six years, from September 2016 to June 2022. for three reservoirs in Uruguay: Baygorria, Bonete, and Palmar. This data was provided via Attila Lengyel from researchers from the Ministry of Environment of Uruguay. The data included information on Chlorophyll-A concentrations, which are commonly used as an indicator of algal blooms, as well as other environmental variables that may be relevant to algal blooms such as turbidity of the water, colored dissolved organic matter (CDOM) and the temperature of the water.

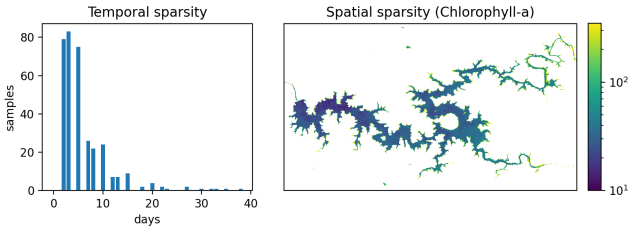


Figure 1: Temporal sparsity of the Chlorophyll-A, colored dissolved organic matter (CDOM) and turbidity bands, and the spatial sparsity of the Chlorophyll-A band, in the Palmar dataset. The bar graph shows the frequency of gaps of a certain number of days in the data. The second image illustrates the number of missing samples at each coordinate. Pixels with fewer or no missing samples are represented by darker shades, while pixels with higher numbers of missing samples are represented by lighter shades.

For this research, the data from the Palmar reservoir was chosen because more data was available in that dataset compared to the other two reservoirs. Only the Chlorophyll-A, turbidity, and CDOM bands were used for this research as these bands are commonly used as indicators of algal blooms and have the most sparsity of the available data.

However, these bands all are temporally sparse and spatially sparse. Both the temporal and spatial sparsity of the Palmar dataset are illustrated in Figure 1.

2 Related Work

2.1 Deep Learning in Remote Sensing Applications

Deep learning has been applied to a variety of remote sensing applications, such as in object detection and land cover classification [4]. The most relevant of these is the application of time-series data for predicting changes.

One of the approaches uses a deep stacking network-based model called Deep-STEP [5] in order to predict the normalised difference vegetation index. Another paper [6] explores the use of a recurrent neural network to predict changes in land cover.

2.2 Chlorophyll-a prediction

Various methods for predicting chlorophyll-a concentrations in coastal waters have already been explored in other locations. One paper [7] uses an (long short-term memory) LSTM model to predict the chlorophyll-a concentration in the East China Sea using Moderate Resolution Imaging Spectroradiometer (MODIS) data. Another paper [8] uses a bidirectional version of the LSTM to predict the chlorophyll-a concentration in the Andaman Sea.

3 Methodology

This study aims to investigate the performance of multiple methods of interpolating spatially and temporally sparse satellite data. The first method will be a lookback algorithm that will be used as a simple baseline. The second method will

be a nearest neighbour interpolation method and the third will be a linear interpolation method. After discussing the interpolation methods, the model that will be used for training will be described. Finally, the loss functions will be discussed.

3.1 Baseline Interpolation

As a baseline for comparing the performance of the experiment, a simple look-back algorithm for interpolating the sparse data was used. This algorithm works by filling in missing data points in the time series by taking the latest non-missing spatial values of the past 30 days of data at each pixel location.

This simple look-back algorithm serves as a decent baseline for comparing other, possibly more complex, experiments fairly. Additionally, it serves as a benchmark for evaluating the performance of the other interpolation methods, as it shows what can be achieved by using a very basic interpolation method.

3.2 Nearest Neighbour Interpolation

The next method that will be used for the interpolation of missing spatial and temporal information is the Nearest Neighbour Interpolation Algorithm. This algorithm interpolates missing temporal and spatial samples between consecutive samples by identifying the closest measured value in the original samples for each value in the newly interpolated samples.

Nearest Neighbour Interpolation offers a significant advantage over Baseline Interpolation as it has the capability to fill in missing spatial information, whereas Baseline Interpolation is limited to only using the previous samples that contain the missing spatial information. Another advantage of this capability is that Nearest Neigbour Interpolation uses current samples, which may be more relevant, to interpolate missing spatial information.

3.3 Linear Interpolation

The Linear Interpolation Algorithm is another method that can be used to interpolate the input data. This algorithm uses a line to interpolate between known values in consecutive samples.

Linear Interpolation offers the same advantage over Baseline Interpolation as Nearest Neigbour Interpolation, in that it has the same capability of interpolating spatial information. Additionally, the Linear Interpolation Algorithm is better at representing the gradients of the spatial and temporal information than the Nearest Neighbour Interpolation Algorithm. This is because the Linear Interpolation Algorithm captures the gradual changes in the original data, which the Nearest Neigbour Interpolation Algorithm is unable to do.

3.4 The model: U-Net

To investigate the performance of each interpolation method, a U-Net [9] model will be used to train on. U-Net is a type of convolutional neural network (CNN) that is commonly used for image segmentation tasks originally introduced in [9]. The architecture (see Figure 2) consists of a contracting path, which downscales all the spatial features, and an expansive path which upscales the spatial features again.

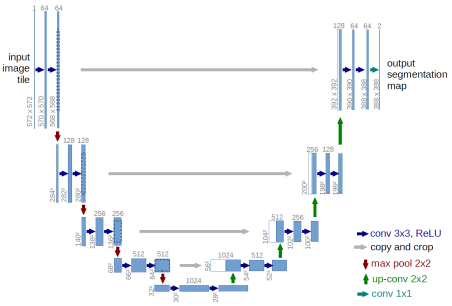


Figure 2: U-Net architecture. [9]

3.5 Loss Functions

In machine learning, a loss function is a function that calculates the difference between the predicted output and the true output. The goal of training a model is to minimize the value of the loss function. There are many different types of loss functions, each with its advantages and disadvantages. In this research, the root mean squared error (RMSE) loss function was used for validation, and two loss functions were used for training: mean squared error (MSE) and DenseWeight.

Mean Squared Error

Mean squared error (MSE) is a commonly used loss function in regression tasks. MSE is defined as the average of the squared differences between the predicted and true values, defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

where Y is a vector of true values, $\hat{Y}$ is a vector of predicted values with the same shape as Y, and n is the total number of values.

RMSE

Root mean squared error (RMSE) is in principle the same metric as the MSE loss, but with the result in the same unit as the input.

$$RMSE = \sqrt{MSE} \tag{2}$$

DenseWeight

DenseWeight [10] is a weighting function that can be used with other loss functions, such as MSE, to handle imbalanced data. Imbalanced data refers to a situation where the distribution of the labels is not uniform. When a model is trained on imbalanced data, it could underperform in the prediction of rare cases.

The data used in this research was also imbalanced, as shown in the density plot of the Chlorophyll-A band in Figure 3. To tackle this problem, DenseWeight is used as a weighting function. It assigns different weights to different labels, with higher weights for rare cases. To achieve this, DenseWeight first uses kernel density estimation (KDE) [11] to estimate the probability function of the data:

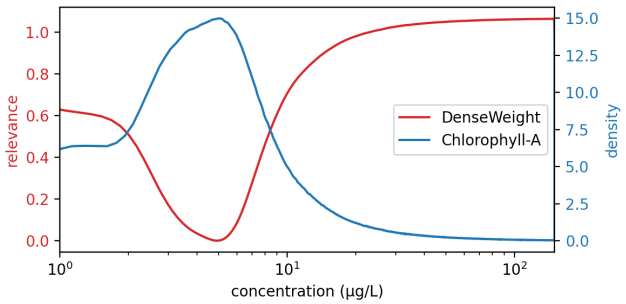


Figure 3: DenseWeight [10] function applied over the Chlorophyll-A band in the Palmar dataset. This figure illustrates the relationship between the density distribution of the Chlorophyll-A band and the relevance computed by the DenseWeight function with $\alpha = 1$.

$$p(y) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{y - Y_i}{h}\right) \tag{3}$$

where $p(y)$ is the estimated probability density function (PDF), Y is the full dataset, n is the total number of values in the dataset, h is the bandwidth, and K is the kernel function. For bandwidth selection, Silverman's rule was used [11]. Next, the estimated PDF is normalised between zero and one:

$$p'(y) = \frac{p(y) - \min(p(Y))}{\max(p(Y)) - \min(p(Y))} \tag{4}$$

After that, the final weighting function is constructed:

$$f_w(\alpha, y) = \frac{\max(1 - \alpha p'(y), \epsilon)}{\frac{1}{n}\sum_{i=1}^{n}(\max(1 - \alpha p'(y_i), \epsilon))} \tag{5}$$

where α is a parameter controlling the weighting, and ϵ is a small constant to prevent divide-by-zero errors in the loss computation. This weighting function is plotted in Figure 3. After establishing the weighting function, this function can be used in conjunction with any other loss function. In this research MSE was used to construct the final loss function:

$$MSE_{DenseWeight}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} f_w(\alpha, Y_i) \cdot (Y_i - \hat{Y}_i)^2 \tag{6}$$

where Y is a vector of true values, $\hat{Y}$ is a vector of predicted values with the same shape as Y, and n is the total number of values.

4 Experiments

4.1 Setup

All the experiments in this research were performed on the High-Performance Computing (DHPC) cluster [12] on GPU nodes equipped with NVIDIA Tesla V100S GPUs. The training code was written using PyTorch [13] and PyTorch Lightning [14]. TorchGeo [15] was used for loading the satellite data, which used the GeoTIFF format to store the daily samples. For logging the intermediary results of the training runs, weights and biases [16] was used.

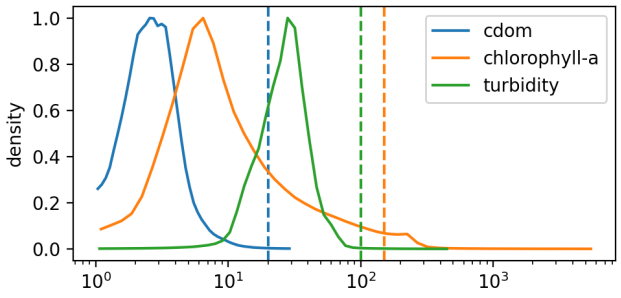


Figure 4: Density of the Chlorophyll-A, colored dissolved organic matter (CDOM), and turbidity bands in the Palmar dataset. The dashed lines show the boundary where values were clipped.

4.2 Experimental Settings

For all experiments, a window size of 5 was used and a learning rate of 0.001. The DenseWeight loss function was used with $\alpha = 1$ and $\epsilon = 1e - 6$. The dataset was split into 80% of training data and 20% of validation data.

4.3 Data Preprocessing

The data used for the experiments is first interpolated with the interpolation method relevant to each experiment. After that, the data is clipped and then normalised.

Clipping is a technique used to remove any data that falls outside of a specified range. This is done to eliminate outliers or extreme values that can skew the results. The clipping values chosen for this research are shown in Figure 4 and were chosen based on preliminary analysis of the data.

After clipping the data, each band was normalised to a range between 0 to 1. This is done to ensure that all the different bands of the dataset are on the same scale and also to ensure that the gradients on the neural network do not explode.

4.4 Baseline

The baseline results for the MSE and DenseWeight loss functions are shown in Figure 5. Examples of these results are

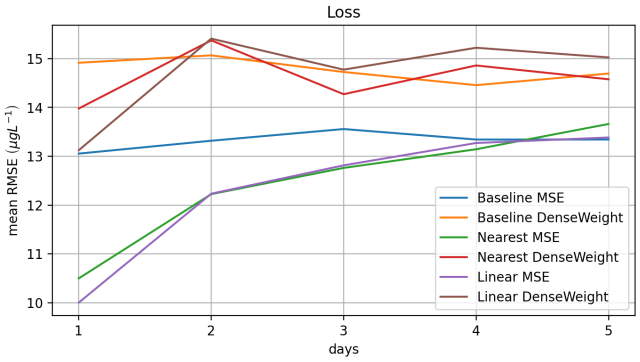


Figure 5: Averaged root mean square error (RMSE) loss for each experiment (n = 3). The horizontal axis shows the prediction horizon in days. The vertical axis shows the mean RMSE in μgL^{-1}. These values are taken from Table 1.

in Figure 6a and Figure 6b. As can be seen in the graph the MSE loss is lower for all used prediction horizons than the DenseWeight loss. The MSE loss seems to only raise slightly until a prediction horizon of 3 after which the MSE loss seems to go down again. The DenseWeight loss only raises after the first increase in the prediction horizon after which it seems to lower until the prediction horizon reaches 4, after which it increases again. The fact that the loss stays high and looking at the results presented in Figure 6a and Figure 6b means that the baseline interpolation method seems to fail in properly predicting the chlorophyll-a concentration for any prediction horizon.

4.5 Nearest

The results of the nearest neighbour interpolation method for the MSE and DenseWeight loss functions are shown in Figure 5. Examples of these results are in Figure 6c and Figure 6d. As can be seen in the graph the MSE loss is lower for all used prediction horizons than the DenseWeight loss. The MSE loss shows an upward trend while the DenseWeight loss only raises after the first increase in the prediction horizon. After that, the DenseWeight loss seems to fluctuate around 14.5. The results here seem to indicate that MSE loss works better in general as a training function than DenseWeight loss, however, from the results shown in Figure 6c and Figure 6d the MSE loss function seems to underpredict the higher intensity chlorophyll-a values while the DenseWeight loss seems to mostly overpredict on the higher intensity chlorophyll-a values.

4.6 Linear

The results of the linear interpolation method for the MSE and DenseWeight loss functions are shown in Figure 5. Examples of these results are in Figure 6e and Figure 6f. In Figure 7 results are shown for both loss functions with a prediction horizon of 1 to 5. As can be seen in the graph the MSE loss is lower for all used prediction horizons than the DenseWeight loss. The MSE loss shows an upward trend while the DenseWeight loss only raises after the first increase in the prediction horizon. After that, the DenseWeight loss seems to fluctuate around 14.5. The results here seem to draw the same conclusion as in subsection 4.5, albeit starting with a slightly lower loss than with the nearest neighbour interpolation method.

5 Responsible Research

An important aspect of research is ensuring that the results can be independently verified and replicated by other researchers. To achieve this, the code used in this research was made publicly available on TU Delft's servers under the GPLv3 license, ensuring that anyone can access, use, and modify the code as long as they share any modifications they make under the same license. This code includes a full specification of all dependencies used, documentation on running data preprocessing and training, and the job scripts used for running the training on a SLURM [17]-supported cluster.

Another important aspect of reproducibility is the use of seed values in the research. Seeding refers to the practice of using a fixed starting point for the random number generators, in order to ensure that results are reproducible across multiple runs. It is important to note, however, that this does not guarantee that the produced results will be identical to those reported in this research[1]. There may be variations in the results due to the difference in hardware, drivers, software, and other factors.

Unfortunately, the preprocessed data used in this research is not publicly available since the algorithms used for removing cloud coverage and estimating the chlorophyll-a concentrations were developed by researchers of the Ministry of Environment of Uruguay. This lack of accessibility of the data may limit the potential for future contributions to this research.

6 Discussion

In comparison to other research [7] the current proposed solution using interpolation performs worse. The results in [7] show that it is possible to accurately predict for up to 3 days and that after that the performance starts to degrade. This performance improvement might be the result of LSTM performing better than UNet.

7 Conclusions and Future Work

In conclusion, this research aimed to investigate the performance of different interpolation methods for predicting chlorophyll-a concentrations in satellite data. The results of the experiments showed that the MSE loss function performed better than the DenseWeight loss function in general, however, both loss functions had their own strengths and weaknesses. The MSE loss function managed to consistently predict the lowest difference but failed in taking into account higher chlorophyll-a concentration. The DenseWeight loss function managed to predict 1 day farther into the future than the MSE loss function but overpredicted on lower labels. The baseline interpolation method failed to properly predict the chlorophyll-a concentration for any prediction horizon, while the nearest neighbour interpolation method and the linear interpolation method performed almost equally well, with the linear method just outperforming the nearest neighbour method.

For future work, it would be interesting to explore the use of different interpolation methods, different deep learning models, and different loss functions to find a better balance between the MSE and the DenseWeight loss functions. It would also be interesting to find which other data modalities would be able to improve the results.

References

[1] Mark L. Wells et al. Harmful algal blooms and climate change: Learning from the past and present to forecast the future. *Harmful Algae*, 49:68–93, 2015.

[2] Darrel L. Williams, Samuel Goward, and Terry Arvidson. Landsat. *Photogrammetric Engineering & Remote Sensing*, 72(10):1171–1178, October 2006.

[1] https://pytorch.org/docs/stable/notes/randomness.html

[3] Philippe Martimort et al. Sentinel-2 optical high resolution mission for gmes operational services. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 2677–2680, 2007.

[4] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 2019.

[5] Monidipa Das and Soumya K. Ghosh. Deep-step: A deep learning approach for spatiotemporal prediction of remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1984 – 1988, 2016. Cited by: 63.

[6] Haobo Lyu, Hui Lu, and Lichao Mou. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sensing*, 8(6), 2016. Cited by: 217; All Open Access, Gold Open Access, Green Open Access.

[7] Haobin Cen, Jiahan Jiang, Guoqing Han, Xiayan Lin, Yu Liu, Xiaoyan Jia, Qiyan Ji, and Bo Li. Applying deep learning in the prediction of chlorophyll-a in the east china sea. *REMOTE SENSING*, 14(21), NOV 2022.

[8] Pawalee Srisuksomwong and Porpattama Hammachukiattikul. The chlorophyll-a modelling over the andaman sea using bi-directional lstm neural network. In *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 955–958, 2022.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[10] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110(8):2187–2211, July 2021.

[11] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[12] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 1). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.

[13] Adam Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

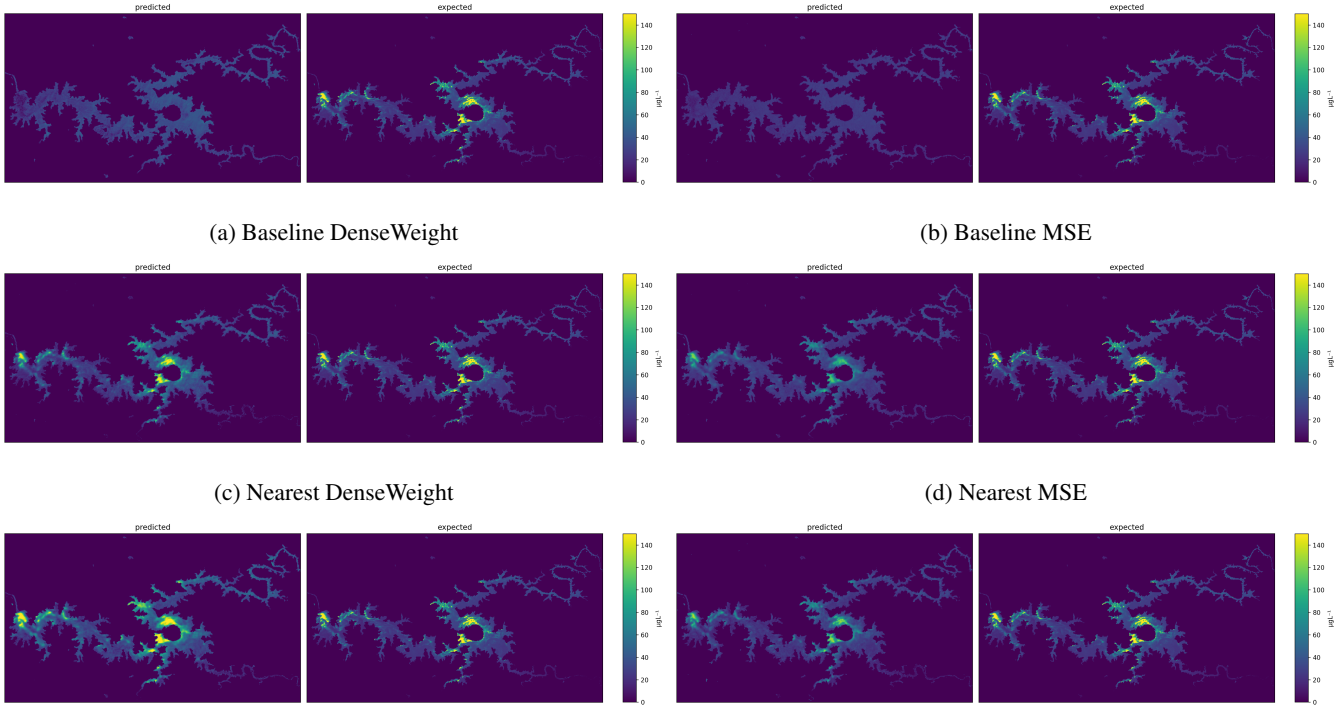
[14] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.

[15] Adam J. Stewart, Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, and Arindam Banerjee. TorchGeo: Deep Learning With Geospatial Data. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, pages 1–12, Seattle, Washington, 11 2022. Association for Computing Machinery.

[16] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[17] Andy B. Yoo, Morris A. Jette, and Mark Grondona. Slurm: Simple linux utility for resource management. In Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 44–60, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

A Results



(a) Baseline DenseWeight

(b) Baseline MSE

(c) Nearest DenseWeight

(d) Nearest MSE

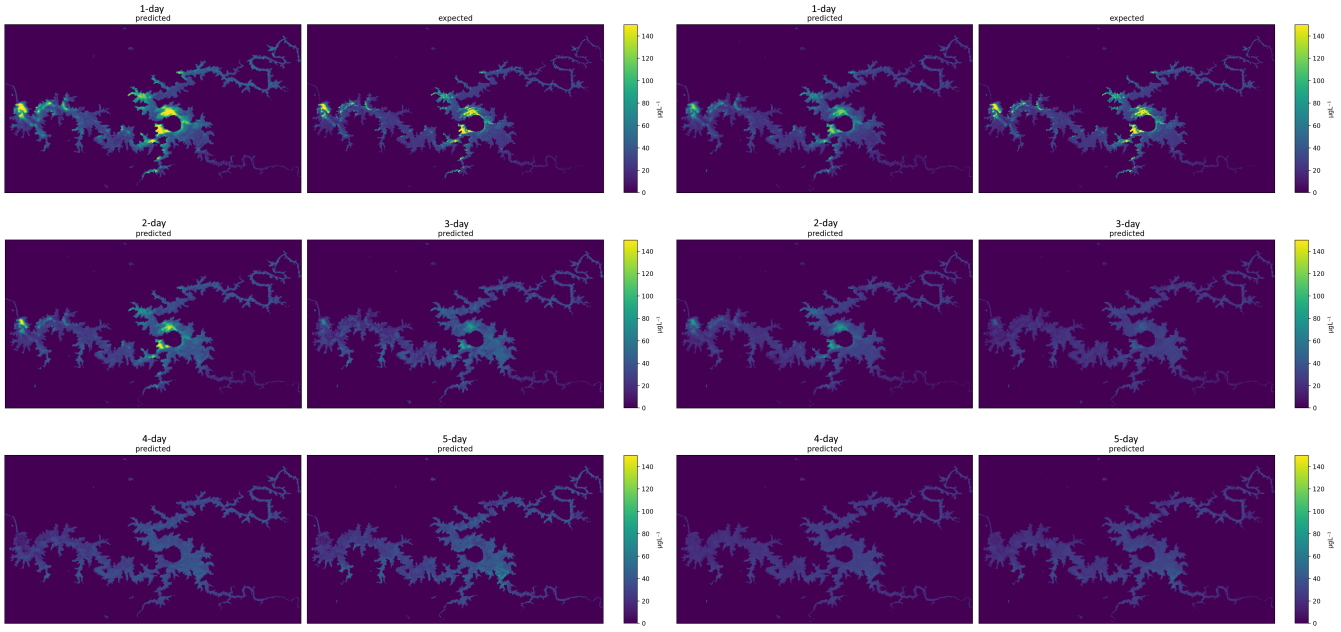
(e) Linear DenseWeight

(f) Linear MSE

Figure 6: Predicted values for each experiment with a prediction horizon of 1.

Interpolation Method	**Loss Function**	**Mean RMSE ± SD** ($\mu g L^{-1}$)				
		1-day	**2-day**	**3-day**	**4-day**	**5-day**
Baseline	MSE	13.05 ± 0.19	13.32 ± 0.31	13.56 ± 0.50	13.34 ± 0.17	13.34 ± 0.46
	DenseWeight	14.91 ± 0.62	15.07 ± 0.70	14.72 ± 0.37	14.46 ± 0.33	14.69 ± 0.48
Nearest Neigbour	MSE	10.50 ± 0.21	12.22 ± 0.19	12.76 ± 0.31	13.14 ± 0.29	13.66 ± 0.36
	DenseWeight	13.98 ± 0.62	15.37 ± 0.68	14.27 ± 0.16	14.86 ± 0.07	14.58 ± 0.15
Linear	MSE	10.00 ± 0.26	12.23 ± 0.09	12.81 ± 0.17	13.27 ± 0.13	13.38 ± 0.19
	DenseWeight	13.12 ± 0.81	15.41 ± 0.97	14.77 ± 0.29	15.22 ± 1.09	15.02 ± 0.94

Table 1: Results. This table compares the mean root mean squared error (RMSE) with the standard deviation (SD) of different interpolation methods (Baseline, Nearest Neighbour, and Linear) using two different loss functions (MSE and DenseWeight) for predicted Chlorophyll-A concentrations with a one- to five-day prediction horizon. The results were obtained by training each configuration three times for 30 epochs with seeds: 42, 43, and 44.



(a) Linear DenseWeight

(b) Linear MSE

Figure 7: Predicted values for each experiment with prediction horizons of 1 to 5. It can be observed that the network correctly predicts higher concentrations of chlorophyll-a concentrations until day 2, or day 3 for the DenseWeight loss, after which the model does not predict any higher concentrations.