Algal Bloom Forecasting

Classical Machine Learning versus Deep-Learning

First 300 words: Abstract & Introduction

Rob Lubbers[1]

Supervisor(s): 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 30, 2023

Name of the student: Rob Lubbers
Final project course: CSE3000 Research Project
Thesis committee: Jan van Gemert, Attila Lengyel, Robert-Jan Bruintjes, Koen Langendoen

Abstract

The aim of this paper is to find out which Machine Learning (ML) model predicts the concentration of Chlorophyll-a, in the Palmar lake in Uruguay best. Currently there are no such models to predict the growth in this lake. The algorithms which will be compared in this paper are a Linear Regression model and the U-Net model. We will compare the losses of the two models to determine which algorithm performs best. The less loss a model has, the more accurate it is, and thus the better it is. The loss of the U-Net model failed to converge to a value, meaning it was impossible to compare the two models.

1 Introduction

In Uruguay there is a need to predict Harmful Algal Blooms (HABs) in their water reservoirs. These HABs pose a threat to the quality of drinking water, outdoor recreation, fish and animals, and human health. The water reservoirs are also used as source of drinking water for local settlements.

Hydroelectric plants have a negative impact on the water quality [18]. These plants cause eutrophication which in turn causes algae to bloom. Because of this negative impact, measurements are already being made of the algae concentration in this lake [8]. However, this data is not yet being used to predict future HABs.

A lot of research has already been done in the field of Machine Learning [4][7][11][12][17]. These models work well in predicting growths[17][13], however none have experimented with these models on this specific lake.

It has been shown that the concentration of Chlorophyll-a correlates with the concentration of algae in a lake [5]. Because there is a need to predict the bloom of algae in this lake, the research question which will be answered in this paper is *how a classical Machine Learning model compares to a more modern Deep-Learning model, in predicting the concentration of Chlorophyll-a in the Palmar lake*.

More specifically, for the more classical model we will be using a Linear Regression model, and for the more modern model we will use the U-Net [9] model. These models will be explained in chapters 3.3 and 3.5 respectively.

In this paper we will first outline in section 2 the general concepts used to approach this research question.

Section 3 describes the metrics by which the accuracy of the models is measured and explain these metrics were chosen. We will also describe the models and set-up we used to solve this problem. Finally, in section 6 we will present the results of the experiment and conclude which model worked best on this problem.

2 Methodology

To answer this research question two models are implemented: a classical Linear Regression model and the U-net architecture as a deep learning model. These models were chosen because the Linear Regression model is a more simple model which can provide a nice baseline. The U-net model has proven to be efficient at classifying satellite imagery [14].

Both models will have a prediction horizon of 1 day, meaning the models will try to predict the chlorophyll-a concentration for the next day. For the input, this means the data of the previous day is used. This was chosen this way to simplify the experiment.

The chosen programming language is Python, because of its vast amount of machine learning libraries available Python has a number of well-established machine learning libraries, including PyTorch. Also its ease of use and interoperability with other platforms, make it a fitting choice.

These machine learning models were implemented with the same Python libraries as to minimize the performance differences which could depend on the libraries used. The library used to create the models is PyTorch, a well-established, specialized library for implementing machine learning models.[1]

There models were ran on the TU Delft super computer, DelftBlue [1], as training these models takes a lot of computing power.

Both models will be trained and validated on the same data sets, in this way we can most accurately compare them. This can be done through seeding. As long as there are no other sources of non-determinism and the same seed is used, the same sets of data will be retrieved.

The models will be trained on 25.000 training samples. We have found that after 25.000 samples the loss doesn't decrease anymore. After every 1% of the samples the performance of the model is evaluated by running it against a test set. Note that the model is not being trained on this test set, that is, the gradients are not being adjusted.

3 Experimental Setup and Results

In this section the experimental set-up is described by describing the different models which are used and how they were implemented.

To be able to compare the accuracy for both models, the same metric has been used to determine the accuracy, namely the Mean Squared Error (MSE) [15]. The MSE is defined as follows:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{1}$$

n is the amount of samples, $\hat{y}_i$ is the predicted value for sample i, and y_i is the actual value (label) of sample i. This way of measuring the error is also used by the model as a criterion to optimize its gradients. We have also looked at the Root Mean Squared Error (RMSE) but we have found that the MSE gave better results, as shown in figure 1.

3.1 Data Preparation

Preprocessing data before it gets fed into a machine learning model has shown to have a positive impact on results [3]. Models do better when the data is less noisy and less irrelevant data is present [6].

[1] https://pytorch.org/

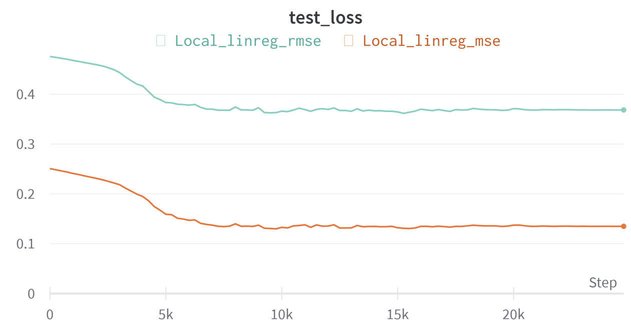

Figure 1: Loss of a linear regression model. The orange line depicts the loss when the MSE is used as a learning criterion and the green line depicts the loss when the RMSE is used as a learning criterion.

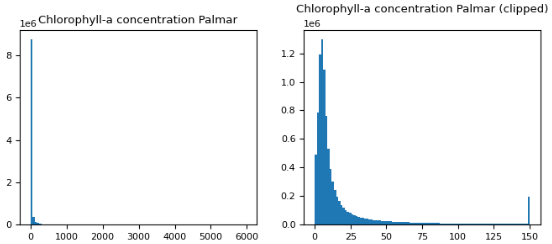

Figure 2: Chlorophyll-a sample distribution, showing why clipping is necessary. The maximum value is 5968.321 whilst most values are between 0 and 50.

Because most of the data came from satellite imagery there are some measurement errors, and a lot of values are Not a Number (NaN) because the data for that pixel is missing. This could be caused by cloud coverage, the satellite not being in the right position to make measurements, or for example it being impossible to measure the water temperature at a pixel which is located on land.

Input values are clipped between $[10^{-6}, 150]$. These bounds were chosen because values over $150\mu L^{-1}$ are unrealistically high, so these are treated as measurement errors. For example, analysis has shown that the maximum value is $5968.321\mu L^{-1}$ which is very unrealistic. 2.046% of not-NaN samples are larger than the threshold of $150\mu L^{-1}$. Figure 2 shows the distribution of chlorophyll-a with and without clipping.

Furthermore, to make the data more normally distributed, a Yeo-Johnsen [16] transformation was applied to the biological and precipitation data bands.

3.2 Data Loader

We have raw data files, however to use these we will first need to access the data. This is done with the help of a dataloader. This dataloader loads the raw files and converts them to usable data. Since loading the entire dataset is impractical, the dataset is about 32 GiB, the dataloader samples random sample instances which can be used for training and testing the models. The dataloader returns a tuple of *(images, masks, targets)* where image is a tuple of shape *(batch size, window size, number of bands, height, width)* where:

- *Batch size* is the amount of batches.
- *Window size* is the amount of sequential samples before the image for which values have to be predicted.
- *Number of bands* is the amount of features.
- *Height, Width* is the height and width of the sample image.

Masks is a tuple containing information on which pixels are unmodified, and *targets* is a tuple containing the labels to be predicted.

3.3 Linear Regression Model

The Linear Regression model has historically been one of the most popular machine learning algorithms [2]. This is greatly due to its simplicity in use and in training.

A linear regression model works by approximating the best linear relationship between the independent variables, the input data modalities, and the dependent variable, chlorophyll-a. It does this by estimating the coefficients of the independent variables in a linear equation, with the goal of minimizing the error between the predicted values and the actual values. To calculate the error the MSE metric was used. Optimizing the gradients is done using the Adam **??** method.

A linear regression model consists of an input layer and an output layer. The input layer consists of nodes, each representing an input value. The output layer would consist of a layer of nodes, where each node would represent the chlorophyll-a value at a pixel. Linear regression models are fully connected, meaning every input node is connected to every output node. However, because of the high dimensionality of our input, this would mean there would be $(batch\_size * window\_size * numberofbands * height * width)$ input nodes and $height * width$ output nodes. If these layers would be fully connected the model could easily become terabytes large, because each connection also stores a weight.

To mitigate this problem, we flatten the input. This means that to predict the value of a pixel y we will only be looking at the value of that same pixel in the images, and not at all other pixels in the image as well. It makes sense to do this, as pixels located far away of the target pixel y will barely have any influence on the value of y. This means that the input layer has size $(batch\_size * window\_size * numberofbands)$ and the output layer has 1 node. We can plot the loss of the linear regression model as a function of the amount of training iterations it has undergone. We can see this plot in figure 3. We find that when running the linear regression model the loss stabilizes around 0.135.

3.4 U-Net Model

In order for the U-Net architecture to process the input data, the data had to be reshaped again. The $window\_size$ and num_bands dimensions are collapsed. An overview of the architecture can be seen in figure 4. The loss of the U-net is plotted in figure 5. We can see that the plot does not converge at a certain value, however note that the model has only undergone 5.000 training iterations. More iterations might improve the loss.

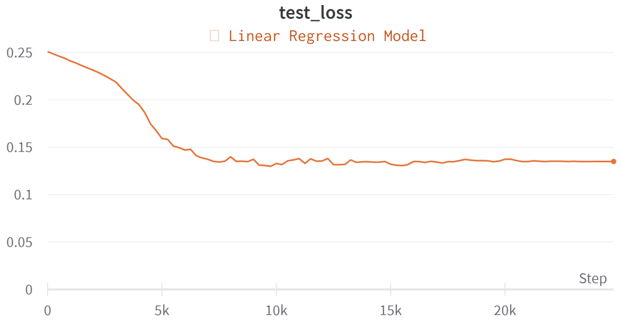

Figure 3: Loss of the Linear Regression model as a function of the amount of training iterations undergone.

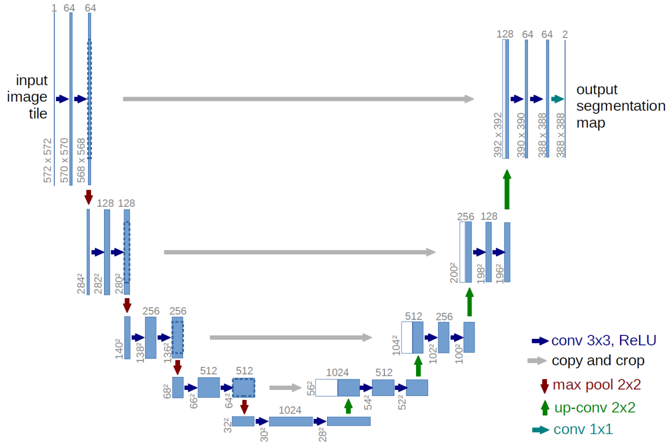

Figure 4: Overview of the U-Net architecture [9].

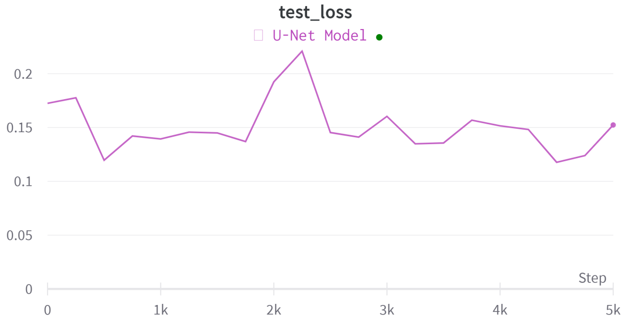

Figure 5: Loss of the U-Net model as a function of the amount of training iterations undergone.

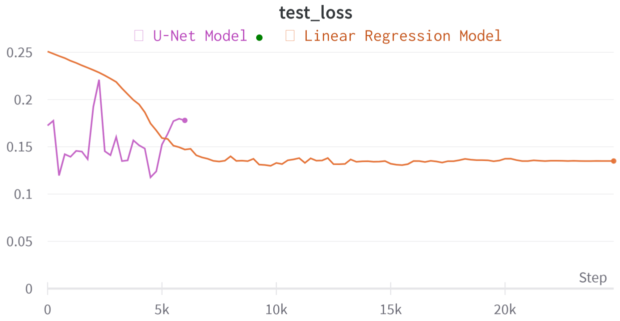

Figure 6: Loss of the U-Net and Linear Regression models as a function of the amount of training iterations undergone.

3.5 Combining the Results

Figure 6 shows the test loss of the Linear Regression model and the U-Net model in a single plot. This plot shows that even though at fewer training iterations the U-Net model has a lower loss than the Linear Regression model, it fails to stabilise at a certain value. The minimal loss of the U-net model is 0.1177, which lower than the stabilised value of the Linear Regression model showing there might be potential for the U-Net model to outperform the Linear Model.

4 Responsible Research

4.1 Ethical Aspects

The data used in this study was provided to us by domain experts, thus no data collection was necessary. This negates the risk of privacy violation. Furthermore, because there was no need for human interaction during the course of this study, there are no ethical risks.

4.2 Reproducibility

The codebase with all the parameters used is available, however not all the datasets are readily available. This makes it hard to reproduce the results which were obtained during this study. However, if the same codebase, data sets and framework were used, one could expect to see similar results. Using a different implementation of the same models can influence the results obtained due to for example the seeding method being different. However, the model like the Linear Regression model should still stabilize at around the same value. Different libraries can also implement different optimisations or implementations which can again also influence the results.

5 Discussion

One limitation of this study is the fact that the U-Net model did not converge. This makes it very hard to draw conclusions from the results we have obtained. Studies have shown great potential in this model though [9], so it's definitely worth researching further.

The reason the model didn't converge is probably not due to dataset issues. This is because the Linear Regression model was trained and tested on the exact same datasets, and it did converge. Most likely the problem lies in implementation issues, either the learning rate needing more adjusting, being too high or too low. Another possibility is simply that the model wasn't given enough time to train. With more iterations it is possible that the model will converge.

6 Conclusions and Future Work

6.1 Conclusions

The Linear Regression model converged nicely to a value, however the U-Net model did not. At several points the U-Net model did have a lower loss than the Linear Regression model had at its lowest point. This means there is potential for the U-Net model to outperform the Linear Regression model, however at this point there is insufficient data to pose a meaningful conclusion. The potential is there for the U-Net model to outperform, these results just didn't show it.

6.2 Future Work

In this paper we have looked at just two Machine Learning algorithms, however there exists a great number of ML algorithms. There might be a different model which could be more accurate than the two researched in this paper. Another model which has shown great promise is for example the ConvLSTM network [10]. Due to the tight time constraints, hardware failing and learning curve involved in this project, there were not a lot of experiments performed. If there were more time, it would be worth looking into implementing this LTSM model and tuning the U-Net model to see if it is possible to make it converge to a value in this setting.

Another recommendation for future work to improve the accuracy of both models is to preprocess the data more. One big limitation of both models is the input data. For example: the input data is temporally and spatially sparse, meaning that at a lot of time frames data is missing, or when there is data at a certain time frame, some areas are missing data. An example of why there is missing data could be cloud coverage - a lot of data comes from satellite measurements. An idea to alleviate this could be to interpolate data points from where there is data, or to take the last known value at a missing data point. In this research all missing points were filled with the value 0.

References

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

[2] Matthias Döring. Supervised learning: Model popularity from past to present, 2018.

[3] Carlos Vladimiro Gonzalez Zelaya. Towards explaining the effects of data preprocessing on machine learning. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 2086–2090, 2019.

[4] 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.

[5] Offoro N. Kimambo, Hector Chikoore, Jabulani R. Gumbo, and Titus A.M. Msagati. Retrospective analysis of chlorophyll-a and its correlation with climate and hydrological variations in mindu dam, morogoro, tanzania. *Heliyon*, 5(11):e02834, 2019.

[6] Sotiris Kotsiantis, Dimitris Kanellopoulos, and P. Pintelas. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1:111–117, 01 2006.

[7] 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.

[8] Mauricio Piana, Daniel Fabián, Andrea Piccardo, and Guillermo Chalar Marquisá. Dynamics of total microcystin lr concentration in three subtropical hydroelectric generation reservoirs in uruguay, south america. *Bulletin of Environmental Contamination and Toxicology*, 99:1–5, 10 2017.

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

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015.

[11] Weilong Song, John M. Dolan, Danelle Cline, and Guangming Xiong. Learning-based algal bloom event recognition for oceanographic decision support system using remote sensing data. *Remote Sensing*, 7(10):13564–13585, 2015.

[12] K. A. Steidinger and K. Haddad. Biologic and Hydrographic Aspects of Red Tides. *BioScience*, 31(11):814–819, 12 1981.

[13] Mukul Tewari, Chandra Kishtawal, Vincent Moriarty, Pallav Ray, Tarkeshwar Singh, Lei Zhang, Lloyd Treinish, and Kushagra Tewari. Improved seasonal prediction of harmful algal blooms using large-scale climate indices. *Communications Earth Environment*, 3, 08 2022.

[14] Priit Ulmas and Innar Liiv. Segmentation of satellite imagery using u-net models for land cover classification. *ArXiv*, abs/2003.02899, 2020.

[15] Weijie Wang and Yanmin Lu. Analysis of the mean absolute error (mae) and the root mean square error (rmse) in assessing rounding model. *IOP Conference Series: Materials Science and Engineering*, 324(1):012049, mar 2018.

[16] In-Kwon Yeo and Richard A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 2000.

[17] Peixuan Yu, Rui Gao, Dezhen Zhang, and Zhi-Ping Liu. Predicting coastal algal blooms with environmental fac-

tors by machine learning methods. *Ecological Indicators*, 123:107334, 2021.

[18] Rahim Zahedi, Abolfazl Ahmadi, and Siavash Gitifar. Reduction of the environmental impacts of the hydropower plant by microalgae cultivation and biodiesel production. *Journal of Environmental Management*, 304:114247, 2022.