Improving Freshwater Quality Measurements through Machine Learning

Jimmy Vuadens
jimmy.vuadens@epfl.ch

Nicolas Riche
nicolas.riche@epfl.ch

Thomas Rivasseau
thomas.rivasseau@epfl.ch

***Abstract*—Nowadays, 71% of the world population has access to drinking water. However, making measurements to test its quality can be difficult and costly. In most part of the world, water stream sampling and measurements of solutes' concentrations cannot be done frequently enough to guarantee a decent water quality at any time. Therefore, engineers try to design machine learning algorithms to predict the behavior of solutes' concentration in water streams, based only on few measurements. Between March 2007 and January 2009, the Centre for Ecology and Hydrology created a one-of-a-kind dataset of measurement of many usable water components in Wales, sampled every seven hours. Our project seeks to explore if it is possible to use this collected data to create machine learning models which can accurately predict ions and dissolved organic carbon concentrations in water. For this we seek to use only sparse sampling, and samples of easily measurable values such as conductivity or water flux.**

I. Introduction, Goal

We worked on a data set of 7 hour interval exhaustive water quality samples. They were taken at three sampling points in Plynlimon, Wales. The three stations where the measurements are made are: UHF (Upper Hafren), LHF (Lower Hafren) and CHR. We will focus only on the UHF values during our project. Our main goal was to predict the concentration of certain ions and dissolved organic carbon (DOC) concentrations using 7 hour interval measurements of:
- water-flux
- PH
- water conductivity

as these values are easily measured. We could also use measurements of our ”target values”, but the goal was to rely on these as little as possible for our predictions. In the real world, measurement of these targets are difficult and costly.

II. Data Exploration, Visualisation

The data is separated in different sets, and we used the ”7-hour edited data”. The raw data has inconsistencies and unusable values. The whole data set is on an excel file, so we used the pandas library to import this data as a dataframe. We first did some visualisations of the major ions we aim to predict:
- Major ions (NO3, SO4, Cl, Na, Mg)
- Dissolved Organic carbon (DOC)

Using these easily measured values:
- PH
- Electric Conductivity (EC)
- Water-flux

The main objective was to get a first intuition that some samples are highly correlated such as Chloride and Sodium that behave similarly to Conductivity.

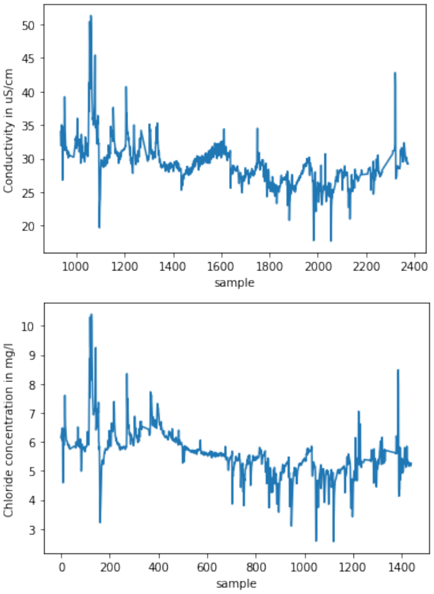

Fig. 1. Graph plots of linearly interpolated Electric Conductivity, truncated Chloride concentration

Electric Conductivity, one of our most important features, has no usable values before sample 935 (04/12/2007). To account for this, we restricted our usable set to samples between 04/12/2007 to 26/01/2009 (end of sampling).
We also noticed that for almost every features, they were some missing values, therefore for each one we used, we filled the blanks with a linear interpolation.
From this point on, we developed our model by focusing on the Chloride concentration, and trying to predict it. The goal is to create a solid model with this target and then extend it to other ions or quantities.

III. Benchmark Values

To compute the efficiency of our model, we compute the absolute mean difference between the exact values of the Chloride concentration and the one calculated by our model.
To evaluate the general efficiency and contribution to the domain of our model we used two benchmarks for

comparison:
- The difference between the linearly interpolated samples of Chloride Concentration every X days and the exact ones.
- The difference between the linear regression using only Electric Conductivity sampled every 7 hours as a feature and the exact samples.

IV. Model-making, Testing

A. First Models

We first needed a model to predict our values. We started with 3 algorithms to compare: a simple linear regression with a gradient descent optimization, a logistic regression using gradient descent with parameters: gamma = 1 x 10^{-9} and 1000 maximum iterations, and finally a ridge regression using the sklearn library of linear models with alpha = 0.1. We only used the three common features that are Conductivity, Water Flux and Ph. As our number of values wasn't consequent (approximately 1400) , we implemented a five-fold cross validation, computing the mean weight and then computing the difference with the exact value.

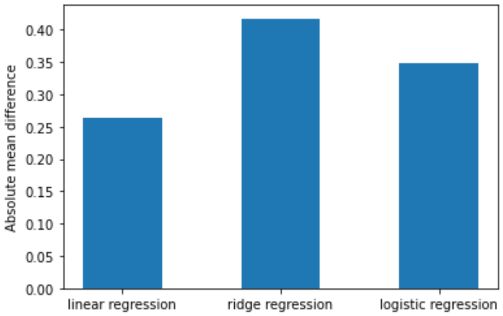

Fig. 2. Difference between the computed value and the exact one for three algorithms

As it was a regression problem, the linear regression algorithm performed the best and we decided to continue with it to improve our features, with 10000 maximum iterations and a gamma of 0.001.

B. Chloride Sampling

Our main goal is to reduce the amount of times we need to measure the desired ion. To do this we integrated as a feature the sampling of Chloride concentration at selected intervals. Fitting a model using samples every 7 hours achieves excellent accuracy but defeats the purpose of this project. The goal is to get an accurate modelling of component concentration using sparse target concentration sampling. We implemented this feature in two different ways:

Repeated Sample :

> We create the feature based on sampling every X days. For the missing values we repeat the measured value until we have another one. This has the benefit that we can instantly compute the next values of the feature when we sample.

Interpolated Sample :

> We create the feature based on sampling every X days. We compute missing values by linear interpolation. This strategy was more precise, but had the major drawback that we need to wait until the new measurement to compute the past values.

We decided on the second solution, as the most important is to reproduce as accurately as possible the missing values.

C. Further Feature Augmentation

Finally, to increase our model accuracy, we added some extra features:

Logarithm of the Water-flux

> This feature is included in the data set, it slightly improves accuracy.

Derivative of the Water-flux

> Meteorological conditions evolve quickly. This helps take these changes into account.

Yearly Sine Wave

> Element concentration in water is very different from season to season (e.g. in summer, the ”dry” period has less solute concentration peaks). To model this we implemented a sine wave with a one-year period.

Previous Chloride Sample

> We add a feature which is simply a time-shifted version of our Chloride-sample feature described in B. This way we take into account not just the last but the two latest Chloride Samples.

D. Results

After implementation, our models achieve better (lower) error averages than linear interpolation of Chloride samples when sampling intervals are 2 days or more. For the ”chloride samples” feature described in part B, interpolation reaches better results than the repetition of the sampled value. This model achieves best accuracy with 0.21 mg/L error when using chloride samples at 1-week and 14-day intervals.

E. Feature Classification

To help with selecting essential features, we provide a table of feature classifications. We hope this will help decide which measurements are most important in the field, especially if resources are limited.

Rank	Feature	Mean relative weight
1	Interpolated Chloride Samples	0.293
2	Water flux	0.276
3	Electric Conductivity	0.157
4	pH	0.116
5	Derivative of the Flux	0.079
6	Previous Chloride Sample	0.033
7	Logarithm of the Flux	0.025
8	Yearly Sine Wave	0.010

TABLE I
Feature Classification by mean relative weight in regression output weights

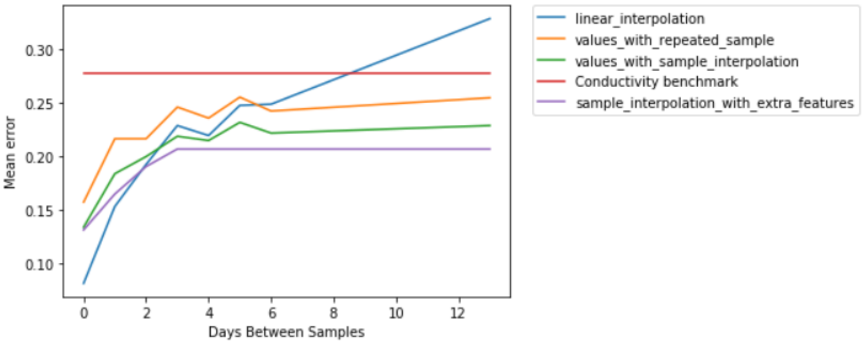

Fig. 3. Mean error of prediction with respect to original Chloride Concentration samples.

V. Neural Network

Using our previous model, the least square gradient descent, we managed to improve our feature vector to get the most precise approximation. But we wanted to see if we could get a better result using another model and this feature vector.

A. *Neural network implementation*

We used the Keras library to implement our Neural Network. We used 7 out of the 8 features as an input dimension, as the yearly sine wave was negligible. Then, we set 3 hidden layers of size 32 ,16 and 8. We used Dense layer as we only need linear operations on our layer's input vector. Contrary to our first model, we now use the Adam optimizer instead of the gradient descent, and use the mean square error to compute the loss. We split our data in 80% for training, and 20% for testing with the sklearn library and the functions train_test_split.

B. *Neural Network result*

We run the neural network many times to get an average difference between our computed value and the exact one, as the result differ according to the values selected to be the test values. The average difference is 0.17, which is better than our previous model, as can be seen in figure 4.

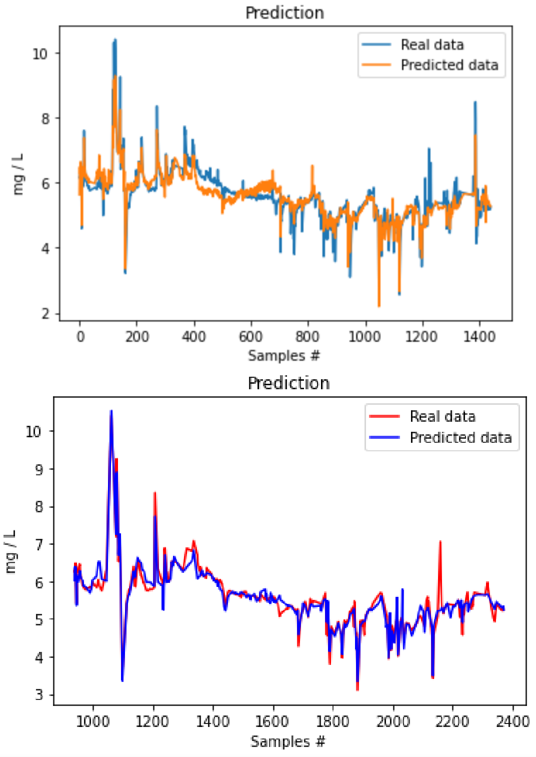

Fig. 4. The predicted value compared to the exact one with Gradient descent and then Neural Network

VI. Extension to other Ions

Now that we had an efficient model with Chloride, we wanted to see how it would perform to predict the behavior of other quantities present in the data set such as Major ions (NO3, SO4, NA, MG) and DOC. To compare performance, since solute concentrations have different scales, we normalised the values with a minimum maximum normalization. This way all values are between 0 and 1, the highest equals 1 and the lowest equals 0. The average relative error between the values computed and the exact values is presented in Fig.5 for each element. We also plot the result of a simple linear interpolation of the ion to compare our result with. The model performs best with Chloride and Sodium. Worst-case performance is for NO3, where we have a relative error of approximately 0.07.

To go further with other ions, we would need to add new features. As an example, the Silicium concentration is high with water stagnation, and low when the water flux is high, due to rainfall for example.

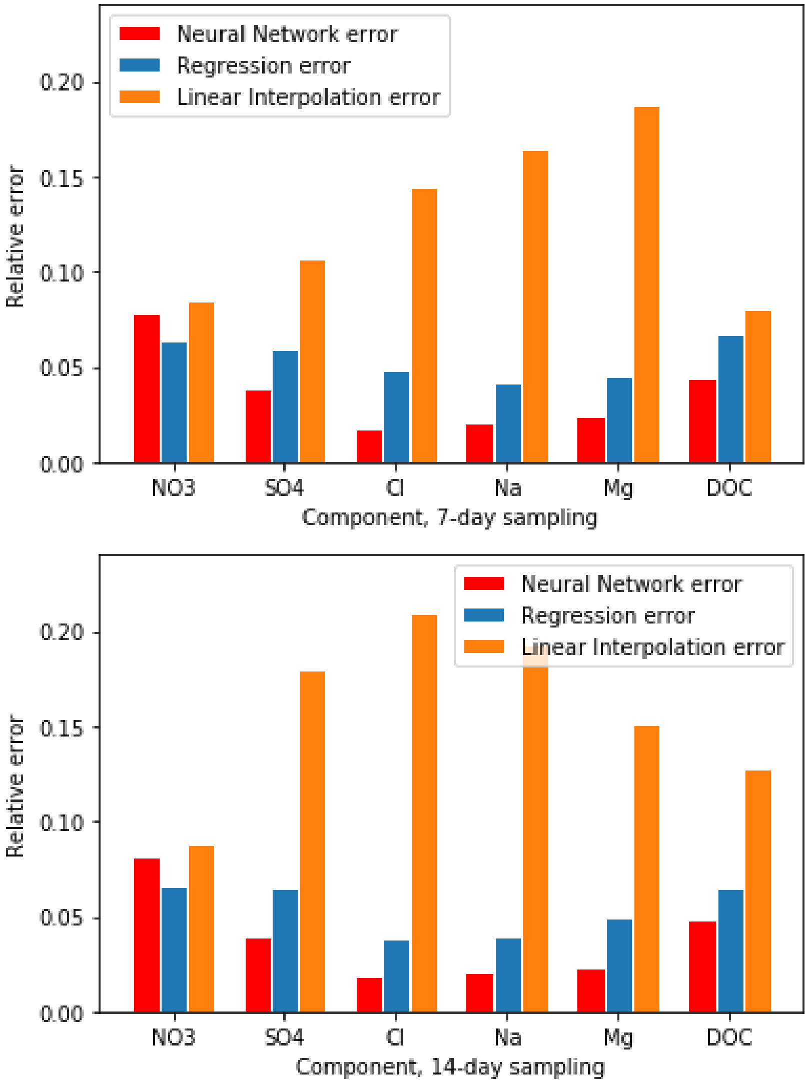

Fig. 5. Predictions output, relative error with respect to sampled values. Sampling intervals of 7 and 14 days.

VII. Acknowledgements

This project was provided by and for the EPFL ECHO lab. We would like to thank Paolo Benettin for his time, guidance, and explanations of the main issues of this project.