A COST-EFFECTIVE VIRTUAL SENSOR FOR CONTINUOUS FRESHWATER NUTRIENT MONITORING USING MACHINE LEARNING

Andrew Zhou[1], Ivan Revilla[2]

[1]Lexington High School, 251 Waltham St, Lexington, MA 02421
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Nutrient enrichment of aquatic environments is a prevalent issue with wide-reaching negative implications for ecological stability, tourism and recreation, and vital drinking supplies. Proper management of nutrient influxes—primarily nitrogen and phosphorus—into aquatic environments is facilitated by continuous monitoring of nutrient levels within water bodies of interest, which offers a more complete understanding of seasonal trends and faster response times compared to traditional lab testing. However, continuous nutrient monitoring systems are prohibitively expensive, with ongoing energy and maintenance requirements that limit deployment. Machine learning shows potential for virtual sensor development with real-time nutrient prediction, based on continuously monitored surrogate indicators. In this study, we test the feasibility of this premise by evaluating the performance of Random Forest regressor (RF), k-Nearest Neighbors (kNN), Support Vector Machine regression (SVM), Decision Tree regressor, Artificial Neural Network, Gradient Boosting Regressor (GBR), and Histogram Gradient Boosting Regressor (HGBR) on one year of water quality testing data from sites across the Continental United States (CONUS). To address values missing not at random, an issue prevalent in water quality testing data, important surrogate indicators are identified by permutation importance. Models are then trained and tuned with Bayesian Optimization to identify hyperparameters optimal for explaining target variance. Across both phosphorus and nitrogen prediction, RF achieved the highest validation performance, with GBR and HGBR trailing marginally. Ensemble tree models appear to be well-suited to continuous nutrient monitoring and may be a cost-efficient solution to greatly supplement the existing high-frequency testing network.

KEYWORDS

Freshwater, Total Phosphorus, Total Nitrogen, Ensemble Tree, Bayesian Optimization

1. INTRODUCTION

The widespread adoption of inorganic fertilizer, which contains high proportions of nitrogen and phosphorus essential to plant growth, is central to achieving current levels of global food production. However, the excessive application of fertilizers, combined with animal manure and human wastewater discharge, result in nutrient enrichment and significant transformation of aquatic environments [1]. Eutrophication from human activities diminishes quality of drinking water, deters recreational activity, and impacts ecosystems worldwide, resulting in an estimated $2.2 billion in annual economic loss in the U.S. alone [2]. Eutrophication is associated with harmful algal blooms (HABs) and hypoxia, posing a risk to animals and humans alike. Total nitrogen and total phosphorus, in addition to water temperature and illumination, are the factors

David C. Wyld et al. (Eds): SIPR, NCWC, BIBC, SOFEA, DSML, ARIA, NLP, CSEN -2023
pp. 143-155, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131912

with the most impact in the development of eutrophic bodies [3]. Phosphorus is generally considered to be the limiting nutrient regulating eutrophication in freshwater environments and is often the target of aquatic research and environmental regulation [4]. Nitrogen enrichment is also worth consideration. Ammonium, the most bioavailable form of nitrogen, regulates functions such as toxin production in cyanobacteria, the primary culprit behind freshwater HABs [4]. Fertilizer pollution is a major non-point source of both phosphorus and nitrogen in aquatic environments. Although nitrate, a highly mobile form of nitrogen in water, is relatively non-toxic, its metabolites, such as nitrite and nitric oxide, pose life-threatening health risks and make nitrates a target for regulation [5][6]. Widespread, continuous monitoring of nutrient concentrations would facilitate faster and better-informed management decisions to mitigate eutrophication, a primarily-anthropogenic issue causing ecological and economic harm.

Three similar studies were found on the application of ML to continuous nutrient monitoring in aquatic environments. Shen et al. [7] also use data from the EPA's Water Quality Portal to train RF models, but process test results from a far longer timeframe to develop seasonal models for total nitrogen and total phosphorus. Despite their expanded dataset, which necessitated multicore supercomputing, the final models developed were in line with or less effective than our model. We also expanded on their approach by testing the performance of six additional models and optimization of hyperparameters to establish the potential of two additional models, GBR and HGBR, in nutrient prediction.

Recently, Paepae et al. compared the feasibility of continuous nutrient monitoring with various ML models, finding that RF and ET performed well on their two selected water bodies [8]. Both rivers they studied had long-term continuous monitoring data available, allowing them to achieve relatively high model accuracies. However, the lack of complete high-frequency data across a majority of testing sites and necessity for individualized model training to each body of water restrict the applicability of their approach. Our evaluation including traditional lab testing data from sites across the CONUS should better account for variation between water bodies to create a more versatile prediction model for deployment across multiple locations.

Elsewhere, Chen et al. tested RF, SVM, and BPNN models for estimation of riverine total phosphorus, total nitrogen, and ammonia across data sampled at different frequencies [9]. Their study similarly found that RF outperformed SVM and BPNN models, as well as finding that higher frequency data improved prediction performance. Notably, this study focuses on testing data from one single automatic testing system, increasing the potential that the performance seen is not representative of all ML nutrient prediction across the region. With a diverse dataset, our evaluation of a wide array of supervised ML models suggests the suitability of RF, GBM, and HGBM across a more representative set of testing sites.

Applying Machine-Learning (ML) to nationwide water quality testing data may offer a robust and cost-efficient method of monitoring concentrations of nitrogen and phosphorus in surface waters. ML algorithms tend to outperform traditional regression in modeling non-linear relations, such as those that relate nutrient concentrations to other often-measured water quality indicators (see Figure 1).

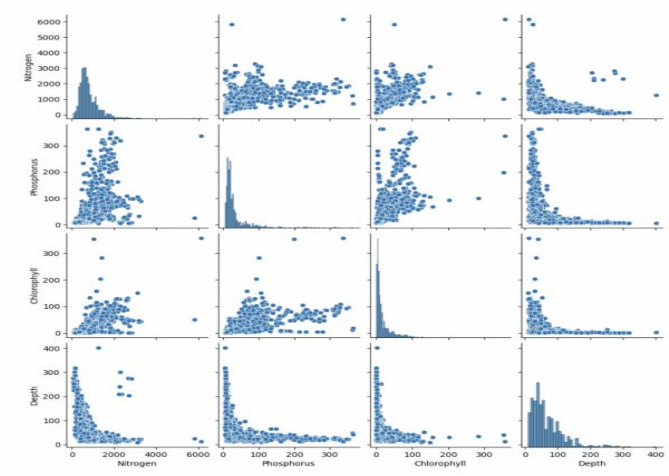

Figure 1. Plot graph showing non-linear correlations between water quality characteristics

High-frequency data, which could be provided with a prediction model and readily-measured characteristics, can better highlight seasonal trends in nutrient concentration, reduce sampling bias, and illustrate aliasing by traditional discrete sampling taken 12-18 times per year [10]. Currently, three major technologies are commercially available for continuous nutrient testing: UV sensors, wet-chemical colorimetric sensors, and ion-selective electrodes (ISE) [11]. For in situ phosphorus testing, only wet-chemical sensors are currently available. Given the high cost of optical UV nitrate sensors ($15,000-$25,000) and wet-chemical nutrient sensors (>$10,000) and frequent calibration requirements of ISE systems, an algorithm-based approach to continuous nutrient monitoring could offer aforementioned insights at significantly reduced costs, enabling greater deployability and a more granular understanding of the impacts of aquatic nutrient contamination.

The dataset utilized in our analysis includes all tests in the EPA Water Quality Portal performed on lake, reservoir, and impoundment sites between July 5, 2022 and July 5, 2023 from sites primarily located within the US [12].

The experiments conducted illustrate the potential of improving “black-box” ML models with automated hyperparameter tuning and the versatility of ML powered virtual sensors across different target characteristics.

Manual hyperparameter tuning is a challenging endeavor that often requires years of experience to efficiently perform. Common automated approaches, such as grid search and random search, tend to be slow and computationally-intensive due to their exhaustive search process. We opted to test Bayesian Optimization, an algorithm that continuously narrows down values being tested based on previous cycles, to efficiently identify optimal hyperparameters. Hyperparameters modifying the fitment of the top four models were run through the optimizer for 100-200 iterations, until coefficients of variation appeared to level off. RF and HGBR experienced the greatest improvement of 14.16% and 9.32% increases in R2, respectively. Optimal hyperparameters for these two models likely boosted generalization ability, as training and validation scores were brought closer in line. Meanwhile, Bayesian Optimization of kNN and GBR saw less significant improvements and no reduction of training scores, suggesting the importance of increasing generalization for ML approaches to water quality monitoring.

The methodology developed in this work was applied to total nitrogen to explore the applicability of the framework outside of phosphorus modeling. Important features for nitrogen prediction

were identified with the same combination of HGBR and RF to facilitate removal of excessive missing values, before the set of seven models evaluated with total phosphorus were applied. Following hyperparameter optimization, all models performed marginally better in nitrogen prediction compared to phosphorus prediction, with RF, GBR, and HGBR standing out as consistently effective models across both nutrients, likely due to the inclusion of ensembling methods that mitigate overfitting of training data.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Data Analysis

The raw data extracted from the EPA's Water Quality Portal individually lists each test result separately, even when testing multiple parameters on the same sample. Performing regression analysis on these parameters requires identifying and grouping tests performed on each sample. One possible approach is to group based on the metadata retrieved on each test, including testing site coordinates and timing. This facilitates establishing correlation between indicators.

2.2. Water Quality Parameters

Testing stations located across the United States test for a variety of different water quality parameters. The data retrieved from July 2022 to July 2023 includes test results for 714 physical, chemical, and biological characteristics, with 790,392 values from 7,723 sites in total. No sample was tested for all characteristics; the sites testing for the maximum of 200-250 characteristics tend to have between 1 and 4 annual visits. To obtain a dataset suitable for regression, which would be used to establish associations between different characteristics, test results for multiple characteristics should originate from near-identical samples. Selecting important characteristics frequently sampled together is critical to maintaining a large sample size, which may mitigate exaggerated accuracy from random chance or overfitting of training data [13].

2.3. Selecting the Correct ML Model

Selecting the correct ML model is fundamental to the accuracy of prediction. Due to the labeled nature of water quality testing indicators, focus was put on supervised learning methods, which attempt to connect and identify relationships between inputs and outputs to generate a model [14]. Many supervised learning models have been developed in recent decades, such as the seven to be tested here: Random Forest regressor (RF), k-Nearest Neighbors (kNN), Support Vector Machine regression (SVM), Decision Tree regressor, Artificial Neural Network, Gradient Boosting Regressor (GBR), and Histogram Gradient Boosting Regressor (HGBR) [15]. Training each of these models offers the option to compare their effectiveness based on coefficients of determination.

3. SOLUTION

The major components in this process include data retrieval, preprocessing, fitting of models, and prediction. One year of water quality data is retrieved from the EPA database using their Tools for Automated Data Analysis (TADA) [12]. This time frame ensures coverage of variation in nutrient concentration over the course of the year, while keeping to a relatively workable size. Attempts to retrieve data on five- or ten-year time spans resulted in timeouts and preprocessing hinderance from memory limitations. The dataset is reformatted through offline processing with

Excel to group concurrent tests on individual samples based on latitude, longitude, and time, as well as converting inconsistent units within individual characteristic measurements. Once grouped, the data are passed through further preprocessing to be standardized and to remove NaNs, which impede model fitting. To fit and optimize models, a train-test split of 0.6 to 0.4 was found to maximize model strength. Following fitting of the ML models, predictions were performed using test data and evaluated.

TADA is a suitable utility for data retrieval due to its accessibility and in-built basic processing functions. As shown in Figure 2, it has the ability to flag and remove nonsensical test values, such as invalid combinations of unit and characteristic, non-numeric values, and duplicates. These entries add complexity to processing and finished models, adding noise that increases the possibility of error in further steps. Thus, the 85,765 flagged test values from 53 sites are removed, reducing the total dataset size to 704,627 values.

Flag reason	Results affected	Required/Optional	Switch 'on' to flag for removal
Sample media is not water	0	Required	
Result value is not numeric or result value is NA and no detection limit value is provided	38082	Required	
Conflict between detection condition text and detection limit type or detection limit type is not in WQX domain tables (likely USGS/NWIS-specific)	255	Required	
Invalid speciation for associated characteristic	17498	Recommended	
Invalid characteristic and fraction combination	1580	Recommended	
Invalid units for a given characteristic and media combination	12983	Recommended	
Measurement activity type code indicates it is a QC replicate, duplicate, or blank	16283	Recommended	
Uncommon analytical methods	1849	Optional	
QAPP approval column is "N" (indicating not approved) or NA	669757	Optional	
Quality Assurance Project Plan URL is missing	784869	Optional	
Metadata indicates result(s) are aggregate, high-frequency continuous sensor results, not discrete samples	0	Optional	
Result value(s) outside of the national upper range for a given characteristic, possibly indicating non-sensical value(s)	10304	Optional	
Result value(s) outside of the national lower range for a given characteristic, possibly indicating non-sensical value(s)	5184	Optional	

Figure 2. Screenshot of TADA processing options

Preprocessing of the retrieved water quality data is central to obtaining a functional prediction model. Over the course of one year, 69,030 discrete samples were taken; in this study, samples taken at distinct latitudes, longitudes, or times are considered discrete. However, no sample was tested for all 713 available characteristics in the database, leading a large portion of the dataset to consist of NaN values. To highlight the scale of this effect, only 9970 discrete samples include measurement of phosphorus, a highly tested nutrient in water and one of the targets of this study. Since many machine-learning algorithms are unable to natively handle datasets with missing values, selecting a smaller subset of characteristics effective in prediction offers an alternative to mass replacement or imputation [8].

```
from sklearn.ensemble import HistGradientBoostingRegressor

model = HistGradientBoostingRegressor()
model.fit(X_train, Y_train)
train_accuracy = model.score(X_train, Y_train)
print(f'The accuracy for the training set is {100 * train_accuracy:.2f}%')
test_accuracy = model.score(X_test, Y_test)
print(f'The accuracy for the test set is {100 * test_accuracy:.2f}%')
```

```
[17] ✓ 0.3s                                                    Python
... The accuracy for the training set is 82.05%
    The accuracy for the test set is 73.66%
```

```
from sklearn.inspection import permutation_importance


#perm_importance = permutation_importance(model, X_test, Y_test)
sorted_idx = np.argsort(perm_importance)
labels = model.feature_names_in_
important_features = pd.Series(perm_importance.importances_mean, index=labels)
important_features.sort_values(ascending=False, inplace=True)
important_features
```

```
[18] ✓ 0.0s                                                    Python
... ORTHOPHOSPHATE                                  0.087278
    NITROGEN, MIXED FORMS (NH3), (NH4), ORGANIC, (NO2) AND (NO3)    0.060323
    CHLOROPHYLL A, UNCORRECTED FOR PHEOPHYTIN       0.254410
    SPECIFIC CONDUCTANCE                            0.093001
    DEPTH, SECCHI DISK DEPTH                        0.073033
                                                        ...
    ZINC                                           -0.000224
    NICKEL                                         -0.000235
    LEAD                                           -0.000253
    URANIUM                                        -0.000369
    BARIUM                                         -0.000378
    Length: 712, dtype: float64
```

Figure 3. Screenshot of code 1

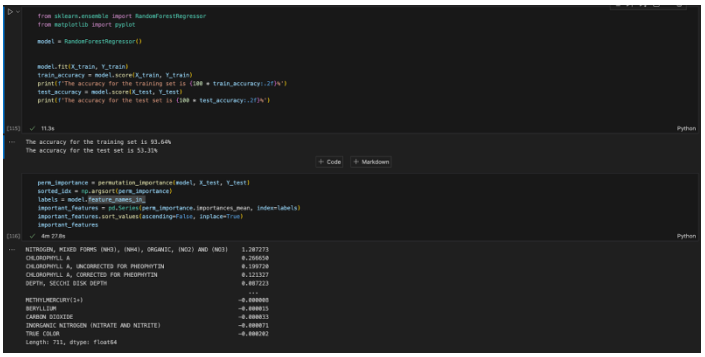

Figure 4. Screenshot of code 2

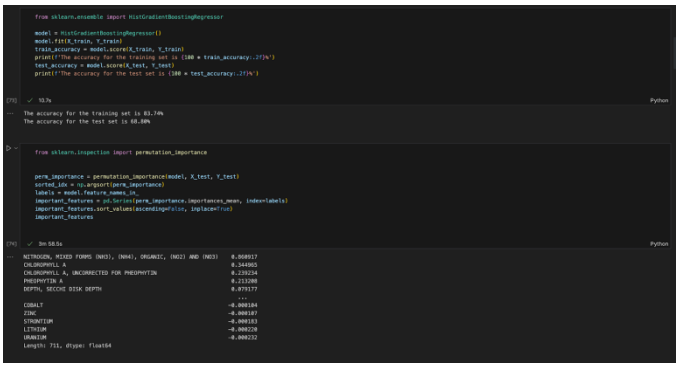

Figure 5. Screenshot of code 3

Since it is unfeasible for every test site to test every sample for all 712 characteristics available in the dataset, there are many values missing not at random. Testing sites likely select characteristics to measure based on their local analytical value; thus, the missing values are neither completely random nor able to be entirely explained by the available data. To effectively select characteristics with useful associations to the prediction target, Hist Gradient Boosting Regressor (HGBR) was applied to the full dataset and permutation importance was calculated (see Figure 3). HGBR is suitable for this initial task due to its support for missing values and relative efficiency on large datasets [16]. As shown in Figure 4, RF was also tested for this application, but required imputation or replacement of missing values; due to the large number of missing values in some characteristics, missing value replacement with zeros was selected over imputation or replacement with median values, in order to minimize influence from generated values.

Initially, the dataset included orthophosphate, a component of total phosphorus representing loose phosphate ions within the water. We chose to omit this characteristic in pursuit of the goal to develop a functional, cost-effective alternative to traditional phosphorus testing. Following the removal of orthophosphate, both models identified, in order of importance, nitrogen, chlorophyll A, and secchi disk depth as features useful in the prediction of phosphorus (see Figures 4 and 5). The HGBR model also identified pheophytin A as an important characteristic.

After identifying important characteristics, a new dataset is created to evaluate ML models less suitable for running on the entire set. Characteristics are selected based on both their initial importance and instances of concurrent measurement with other important features, in order to obtain a large sample of data useful to prediction. Then, individual models are fitted and tweaked to maximize their representation of the relationship between input and target variables.

```
#Splitting into train and test sets
X = df_final.loc[:,df_final.columns != 'Phosphorus']
y = df_final['Phosphorus']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state = 1)

#Random Forest
model2 = RandomForestRegressor()
model2.fit(X_train, y_train)
cross_val_scores2 = cross_val_score(model2, X_train, y_train, cv = cv)
train_accuracy2 = model2.score(X_train, y_train)
test_accuracy2 = model2.score(X_test, y_test)
```

Figure 6. Screenshot of code 4

Creating a useful subset makes training and testing models more efficient in terms of time and computational intensity, while also enabling the removal of missing values to ensure compatibility. However, the set of samples tested for phosphorus, nitrogen, chlorophyll A, uncorrected chlorophyll A, pheophytin A, and secchi disk depth is empty. Between chlorophyll A and uncorrected chlorophyll A, it was found that uncorrected chlorophyll A was measured significantly more often with nitrogen, phosphorus, and secchi disk depth, at 126 and 2014 samples respectively. Pheophytin A is similarly identified as restricting the sample size, having been tested in only 8 samples together with the aforementioned characteristics. To maintain a relatively large sample size, the final subset includes nitrogen, phosphorus, uncorrected chlorophyll, and secchi disk depth.

Using this new subset of the data, various supervised ML models were fitted (see Figure 6). This component involves dividing the new dataset into train and test sets, in order to train the model and then effectively evaluate its accuracy. Without a train-test split, ML models are prone to overfitting to the dataset, resulting in high accuracy with the specific training set but low accuracy when applied to alternate data, such as during utilization of trained models. The test size was adjusted to maximize accuracy on the test set, which serves as an approximation of the entire population. A higher test set accuracy suggests that the model is more effective in representing the general relationship between inputs and outputs and better able to make accurate predictions. After establishing an optimal train-test split of 0.6 to 0.4, we pass through the “X-train” and “y-train” subsets to fit the models.

To establish a baseline, linear regression was performed. The simplicity and prevalence of this model for addressing continuous prediction problems makes it a suitable tool to compare more complex machine-learning models. For the train and test sets, it achieved R2 values of 0.425 and 0.441, respectively.

The fitment and evaluation of the supervised models is best illustrated by this cell. In this example, the model is Scikit-learn’s random forest, an ensemble model that generates multiple different decision trees, called estimators, based on random subsets of the data. This randomization means that not all of the features are made available to each tree, reducing overfitting and ensuring that the estimators are not overly correlated with one another [17]. The model is fit to the training sets of inputs and outputs, before being evaluated with the model.score method, which returns the R2 value of the model when applied to provided datasets. Using default hyperparameters, RF achieved an R2 of 0.939 and 0.565 for train and validation sets, respectively. While modeling more of the target variable variation than linear regression, the significant difference in model effectiveness between train and test data suggests that overfitting of the training data is occurring; the model is picking up patterns in the noise present in the training set, making it less generalized and less able to make accurate predictions with alternate data.

Using the same train-test split, six other models were also trained: kNN, decision tree, SVM, GBR, HGBR, and a sequential neural network. RF, GBR, and kNN were found to be the most effective of the seven models tested. GBR matches the R2 value achieved by RF, with values of 0.769 and 0.593 for train and validation sets, respectively. While RF uses bagging, GBR is based on the boosting method of ensembling, where learners are trained sequentially off previous trees to gradually reduce errors and converge towards an optimal model. HGBR was also tested as a more efficient implementation of this process than traditional GBR because it incorporates discrete binning of continuous data to significantly reduce possible locations to split the underlying tree structure [13]. With this dataset and default hyperparameters, HGBR nears the effectiveness of GBR, with R2 values of 0.798 and 0.558, suggesting that it may be a suitable alternative for larger datasets where the computational intensity of GBR may be impractical or cost-inefficient.

Compared to the ensemble tree algorithms, the basic decision tree and SVM seem less suited to phosphorus prediction. SVM significantly underperforms compared to linear regression, with R2 values of 0.250 and 0.283 for train and test sets, respectively. Decision tree regressor shows strong signs of overfitting, with a training R2 of 0.999 and validation R2 of -0.001; the model is fit so tightly to the training data that it predicts none of the variation present in the test set. While pre-pruning and post-pruning of the tree may be able to significantly reduce this effect, ensemble methods are likely better options overall.

Unoptimized, KNN marginally outperforms the benchmark, with training and test R2 values of 0.627 and 0.477 compared to 0.425 and 0.441 for linear regression. For regression, kNN takes the k closest points to the target to interpolate an output, weighing points uniformly, by distance, or with customized sets of weights. After dimensionality reduction—selecting only the water quality characteristics with greatest impact on the target—the major downsides of kNN are largely addressed; namely, its computational intensity and difficulty with calculating distance on datasets with high dimensionality, also known as the "dimensionality curse" [18]. Thus, it is a viable option following such processing, albeit with a relatively low effectiveness.

4. EXPERIMENT

4.1. Experiment 1

While the Scikit-learn library offers workable default parameters for each ML model, adjusting hyperparameters can improve prediction accuracy or increase fitting efficiency. The initial fitment of the five models we tested took between 10 seconds and 2 minutes running on a portable laptop computer, so we prioritized optimizing prediction accuracy. With larger datasets, computational load may merit additional consideration in order to minimize solution cost.

There are multiple approaches for optimizing machine-learning hyperparameters, the simplest being random search and grid search, which is also most popular [12]. Essentially, a ML model is trained repeatedly using different combinations of hyperparameters until a combination that maximizes a predefined scoring method is found. Grid search runs through lists of user-defined options for each hyperparameter, while random-search randomly generates values for each parameter within user-defined ranges. Bayesian optimization is an alternate method to optimization that recursively builds on information derived from previous training cycles to arrive at optimal hyperparameters in a small number of cycles. As one of the more efficient processes for improving black-box models, it serves our goal to develop an accurate and cost-efficient solution to phosphorus prediction. Bayesian optimization was applied to each of the four models that outperformed the baseline. We opted to maintain the same R2 scoring for the

Bayesian optimizer as our comparison of models with default hyperparameters, in order to maintain comparability between optimized and unoptimized models.

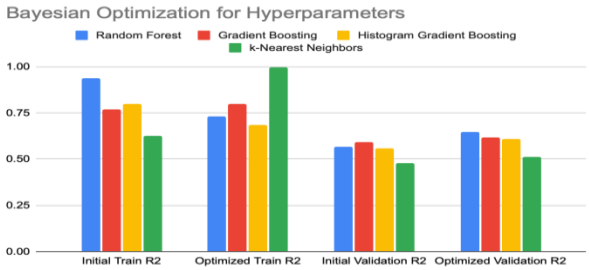

Figure 7. Bayesian Optimization for Hyperparameters

Table 1. Results of Bayesian Optimization in experiment 1

Bayesian Optimization for Hyperparameters				
Model	Random Forest	Gradient Boosting	Histogram Gradient Boosting	k-Nearest Neighbors
Initial Train R2	0.939	0.769	0.798	0.627
Optimized Train R2	0.732	0.8	0.684	0.999
Initial Validation R2	0.565	0.593	0.558	0.477
Optimized Validation R2	0.645	0.619	0.61	0.51

Across the board, applying Bayesian Optimization resulted in higher test R2 values, suggesting that the models were better tuned to the actual relationships between characteristics or less influenced by noise in the training set (see Figure 7 and Table 1). RF had the most significant improvement, with a 14.16% improvement in validation R2. The training set R2 also decreased by 28.27%, bringing the training R2 roughly in line with the validation R2. The optimal hyperparameters involved setting a max depth, the number of decision nodes in each of the estimators, to 235. It also increased the number of underlying estimators from 100 to 118 and set minimums for splits and leaves at 0.00566 and 0.00106, respectively. By imposing limits on the depth of each tree, the optimized RF model is more generalized, as illustrated by the convergence of training and test R2 values.

Bayesian optimization appears to have had a similar effect with HGBR, which experienced a 9.32% improvement in validation R2 and 14.28% decrease in training R2. For HGBR, max_bins and max_iter were the hyperparameters altered most. The max bins parameter, which controls the number of discrete groups the continuous data is separated into, was reduced from the default 255 to 103. The max number of iterations was effectively halved, going from 100 to 51 in the optimized model. These two changes make sense for increasing generalization because reducing the number of bins results in larger individual bins, which serves to ‘blur’ the actual values of the training set. Reducing the number of models in the training sequence reduces opportunity for the model to pick up on noise, resulting in a model better suited for general application.

Gradient Boosting Regressor and k-Nearest Neighbors saw less dramatic increases in validation model strength of 4.384% and 6.918%, respectively. Both also saw increases in training R2, with kNN achieving a training R2 of 0.999. This can be attributed to a change in weighing of nearby points from uniform to distance, which inversely values each neighbor based on their distance and causes the model to tightly fit to every single point in the training set. In theory, it should increase overfitting and make a less generalized model; however, it resulted in a marginal increase in validation R2 as well.

4.2. Experiment 2

While the main focus of this paper is on phosphorus prediction, this framework can be applied to the prediction of almost any characteristic. Take nitrogen, one of the most important features in phosphorus prediction. By setting nitrogen as the target characteristic, the effectiveness of Bayesian optimized machine learning models can be efficiently evaluated, offering insight into the consistency of models across different water quality prediction tasks.

HGBR and RF were run on the entire dataset with nitrogen as the target, in accordance with the procedure previously outlined, resulting in the same four characteristics being identified as important features. This makes sense because nitrogen and phosphorus are tightly correlated as main components of crop fertilizer, which enters freshwater environments and increases levels of turbidity and chlorophyll a. Thus, the turbidity and chlorophyll a relationship to phosphorus also applies to nitrogen. Following selection of important features, each of the seven models was trained and tested. Then, Bayesian optimization is applied to the top four models before comparison. This test is set up through the same pipeline to test the versatility of this process across modeling varying characteristics.

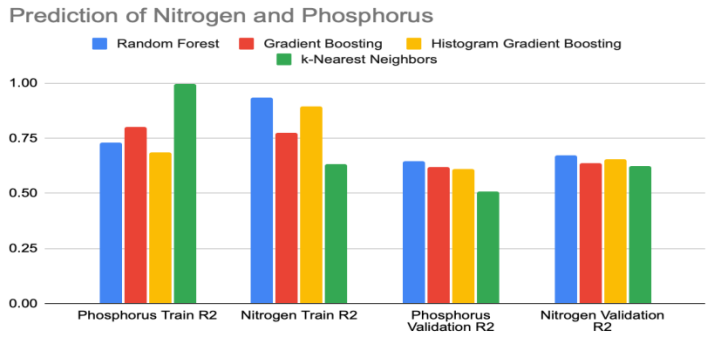

Figure 8. Prediction of Nitrogen and Phosphorus

The top four models in phosphorus prediction: RF, GBR, HGBR, and kNN remain the most effective when applied to nitrogen prediction. Overall, all models predict nitrogen slightly more effectively than phosphorus, suggesting that the relationship between nitrogen and the three input variables is slightly clearer than that of phosphorus (see Figure 8). RF, GBR, and HGBR model approximately the same proportion of variation in the target, with RF retaining the highest validation R2 value at 0.673, indicating their suitability for usage in predictive nutrient monitoring (see Table 2).

Table 2. Model performances across two target nutrients

	Random Forest	Gradient Boosting	Histogram Gradient Boosting	k-Nearest Neighbors
Phosphorus Train R2	0.732	0.8	0.684	0.999
Nitrogen Train R2	0.933	0.773	0.893	0.633
Phosphorus Validation R2	0.645	0.619	0.61	0.51
Nitrogen Validation R2	0.673	0.638	0.657	0.624

5. RELATED WORK

Research by Shen et al. tackles total phosphorus and total nitrogen prediction with a similar approach [7]. Using water quality data derived from the period between 1994 and 2018, they trained RF models to construct a gridded network of phosphorus and nitrogen concentrations across the continental US with a resolution of approximately 1 km, separated by season. Their model tends to achieve lower effectiveness in representing the relationship between input and target variables. For total phosphorus prediction, they achieved a maximum R2 of 0.656 for Spring predictions, with R2 values of 0.533, 0.410, and 0.314 for Summer, Autumn, and Winter predictions, respectively. With our implementation of RF, we achieved an R2 of 0.645 across data from all seasons. Their research processes 47 predictors across a far longer timescale; however, the high intensity, big data approach taken to create their model significantly increases the cost of the implementation for little improvement in accuracy. With the availability of cloud supercomputing, computational intensity can be directly related to cost of implementation; thus, building a smaller model may be more sensible, particularly without decrease in prediction capability.

Another paper by Paepae et al. approaches nutrient prediction with a variety of different supervised ML models, including RF, kNN, and extremely randomized trees (ET) [8]. The team takes a very similar approach to our research, retrieving UK publicly available water quality data, processing it with scalers, and training models with grid search optimization to develop a virtual sensor for nitrogen and phosphorus concentration. Notably, they train separate models on 2-3 years of continuous monitoring data from the two water bodies studied. They found that RF and ET were both extremely effective at predicting total phosphorus, achieving max R2 values of 0.8059 and 0.8229 in the river 'The Cut' and 0.9386 and 0.9498 in the River Enborne. However, the river-by-river approach to their modeling makes it highly unlikely that either model has generalization ability to other bodies of water, particularly with their usage of continuous testing data. Relatively few bodies of water are equipped with a whole range of continuous monitoring equipment, limiting the potential range of application.

A third study on ML estimation of total nitrogen and total phosphorus concentration was used to evaluate the impact of data input frequency in the Chinese Aitoutan watershed [9]. The researchers tested three models: RF, SVM, and back-propagation neural network on 4-hourly, daily, and weekly data across a period of approximately two years. This paper restricted inputs to five water quality indicators cost-effectively continuously measured. As with our analysis, they found that turbidity (measured as secchi disk depth in our dataset) was strongly positively correlated with both nitrogen and phosphorus. RF was also identified as the most effective model with the highest coefficient of determination, achieving an R2 of 0.602 at the weekly timeframe. This study indicates that high-frequency monitoring data significantly increases the effectiveness of nutrient prediction, with the RF model accounting for 78.5% of phosphorus variation with four-hourly data. Unlike this study, our dataset primarily consists of low frequency lab test data from freshwater bodies across the entire CONUS, including significantly more variation; even so, when compared to this study's closest parallel of the weekly timeframe data, the importance of hyperparameter optimization is apparent, as our more versatile model achieves a 7.14% higher R2. Incorporation of more sources of high-frequency testing data may result in further increases in model aptitude.

6. CONCLUSIONS

While the feasibility of a small-scale nutrient prediction model has been evaluated with actual water quality data, there are potential limitations and further research directions to be considered.

In the interest of developing a robust model for nutrient prediction across the entire Continental United States (CONUS), water quality testing data was retrieved from all testing stations. However, the heterogeneous distribution of testing stations across different regions and spatial variation of nutrient pollution may result in differing performance of proposed models across different regions. The testing sites incorporated into the final dataset were primarily selected for consistent simultaneous testing of important characteristics, rather than a representative sample of all freshwater bodies in the CONUS. Thus, for the future, we can try to randomly select testing sites across various strata, such as region, body size, or eutrophication level. Additionally, our feature identification and listwise deletion approach to missing values effectively cleaned the dataset, but introduces a high likelihood of sampling bias. Although related studies have found that imputation methods for missing value replacement fail to improve nutrient prediction, further exploration in advanced imputation may enable the consideration of more well-correlated characteristics that are tested together with phosphorus and nitrogen less frequently.

Across total nitrogen and total phosphorus prediction, the RF, GBR, and HGBR ensemble trees show the most potential as a low-cost supplement to infrequent lab testing or expensive nutrient monitoring systems. Combined with off-the-shelf sensors for continuous measurement of surrogate characteristics, an Internet-of-Things approach should be investigated to lower the barrier to entry of nutrient monitoring. Expanding the availability of real-time data in water bodies of lesser criticality or economically-constrained regions would enable faster, better informed water management decisions to more effectively mitigate the impacts of human-driven eutrophication.

REFERENCES

[1] Richardson, Katherine, and Bo Barker Jørgensen. “Eutrophication: Definition, History and Effects.” Eutrophication in Coastal Marine Ecosystems, 1996, 1–19.

[2] Dodds, Walter K., Wes W. Bouska, Jeffrey L. Eitzmann, Tyler J. Pilger, Kristen L. Pitts, Alyssa J. Riley, Joshua T. Schloesser, and Darren J. Thornbrugh. “Eutrophication of U.S. Freshwaters: Analysis of Potential Economic Damages.” Environmental Science & Technology 43, no. 1 (2008): 12–19.

[3] Yu, Jiabin, Zhaoyang Wang, Xiaoyi Wang, Jiping Xu, and Jie Jia. “Study on Mechanism Experiments and Evaluation Methods for Water Eutrophication.” Journal of Chemistry 2017 (2017)

[4] McCarthy, Mark J., Wayne S. Gardner, Moritz F. Lehmann, and David F. Bird. “Implications of Water Column Ammonium Uptake and Regeneration for the Nitrogen Budget in Temperate, Eutrophic Missisquoi Bay, Lake Champlain (Canada/USA).” Hydrobiologia 718, no. 1 (2013): 173–88.

[5] Killpack, Scott C., and Daryl Buchholz. "Nitrogen in the Environment: Nitrogen Cycle." University of Missouri Extension | MU Extension, November 2022.

[6] Shaban, Jehad, Husam Al-Najar, Kumsal Kocadal, Khaled Almghari, and Sahan Saygi. “The Effect of Nitrate-Contaminated Drinking Water and Vegetables on the Prevalence of Acquired Methemoglobinemia in Beit Lahia City in Palestine.” Water 15, no. 11 (2023): 1989.

[7] Shen, Longzhu Q., Giuseppe Amatulli, Tushar Sethi, Peter Raymond, and Sami Domisch. “Estimating Nitrogen and Phosphorus Concentrations in Streams and Rivers, within a Machine Learning Framework.” Scientific Data 7, no. 1 (2020).

[8] Paepae, Thulane, Pitshou N. Bokoro, and Kyandoghere Kyamakya. "A virtual sensing concept for Nitrogen and Phosphorus monitoring using machine learning techniques." Sensors 22.19 (2022): 7338.

[9] Chen, Shengyue, Zhenyu Zhang, Juanjuan Lin, and Jinliang Huang. “Machine Learning-Based Estimation of Riverine Nutrient Concentrations and Associated Uncertainties Caused by Sampling Frequencies.” PLOS ONE 17, no. 7 (2022).

[10] Pellerin, Brian A., Brian A. Bergamaschi, Bryan D. Downing, John Franco Saraceno, Jessica D. Garrett, and Lisa D. Olsen. “Optical Techniques for the Determination of Nitrate in Environmental

Waters: Guidelines for Instrument Selection, Operation, Deployment, Maintenance, Quality Assurance, and Data Reporting.” Techniques and Methods, 2013.

[11] Pellerin, Brian A., Beth A. Stauffer, Dwane A. Young, Daniel J. Sullivan, Suzanne B. Bricker, Mark R. Walbridge, Gerard A. Clyde, and Denice M. Shaw. “Emerging Tools for Continuous Nutrient Monitoring Networks: Sensors Advancing Science and Water Resources Protection.” JAWRA Journal of the American Water Resources Association52, no. 4 (2016): 993–1008.

[12] Smith, Richard A., Gregory E. Schwarz, and Richard B. Alexander. "Regional interpretation of water-quality monitoring data." Water resources research 33.12 (1997): 2781-2798.

[13] Rajput, Daniyal, Wei-Jen Wang, and Chun-Chuan Chen. "Evaluation of a decided sample size in machine learning applications." BMC bioinformatics 24.1 (2023): 48.

[14] Zhu, Mengyuan, Jiawei Wang, Xiao Yang, Yu Zhang, Linyu Zhang, Hongqiang Ren, Bing Wu, and Lin Ye. “A Review of the Application of Machine Learning in Water Quality Evaluation.” Eco-Environment & Health 1, no. 2 (2022): 107–16.

[15] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[16] Kramer, Oliver, and Oliver Kramer. "Scikit-learn." Machine learning for evolution strategies (2016): 45-53.

[17] Cutler, Adele, D. Richard Cutler, and John R. Stevens. "Random forests." Ensemble machine learning: Methods and applications (2012): 157-175.

[18] Kouiroukidis, Nikolaos, and Georgios Evangelidis. "The effects of dimensionality curse in high dimensional knn search." 2011 15th Panhellenic Conference on Informatics. IEEE, 2011.