A Comprehensive Study of Machine Learning Models for the Prediction of pH in Fairfax County

Abstract

Water quality is drastically declining as bodies of water become more and more acidic due to global warming. Because of this, it is crucial to understand what factors determine pH, and how to predict pH values. In this study, we determine which features are most important in determining pH, and also what combination of features and machine learning models give us the best accuracy in the prediction of pH values. We demonstrate that pH values from the previous year and the percentage of rehabilitated sanitary sewer lines are the most important features in determining pH. We also demonstrate that Decision Tree Models and Random Forest Models perform the best across all features and feature subsets.

1. Introduction

Water is an indispensable resource on our planet, and its significance cannot be overstated. It is essential for sustaining life and is also key to the efficient functioning of ecosystems, agriculture, and industry. Even though it is of utmost importance, water quality is quickly declining. As the climate crisis continues to exacerbate, more and more carbon is being released into the atmosphere. A large majority of this carbon is then absorbed by bodies of water, leading to several adverse consequences. The CO2 acidifies the water in a process that is known as acidification. This not only promotes algae blooms but can also release toxic metals from soils into the water. What's worse, the population of acid-sensitive organisms are declining, simply because they cannot survive outside a certain pH threshold. The Chesapeake Bay area in Virginia is one of the largest estuaries in the USA and houses a wide range of aquatic species. Currently, it is already suffering from massive algae blooms that are greatly harming the ecosystem. Luckily, it has not yet succumbed to massive drops in pH yet due to brackish water conditions and dissolving calcite minerals within the Bay. However, if drastic climate change continues to occur, it will soon become inhabitable for all aquatic species, and the water quality will decline past the point of no return. For these reasons, understanding factors that play a role in pH, a measure of the basicity or acidity of water, will help us take measures to protect the Chesapeake. What's more, if we can accurately predict pH based on a few features, this will remove the cost associated with pH sensors and protect against possible inaccurate recordings due to faulty sensors.

2. Background

Most of the research done on pH prediction through machine learning has to do with the prediction of various water quality indexes that contain pH within them. **[1]** employs Random Forest, Neural Network, Multinomial Logistic Regression, Bagged Tree Model, and Support Vector Machine models to predict WQI. It uses pH, Dissolved Oxygen, Biological Oxygen Demand, Electrical Conductivity, Nitratenan N+ Nitritenann, and Total coliform as features. All models performed exceptionally, but MLR and RF gave the highest accuracies, which were 98.99 and 99.83 respectively. **[2]** utilizes Support Vector Machine, Neural Network, Deep Neural Network, and K Nearest Neighbors models for prediction. It finds the greatest success with the backpropagating DNN model, with an accuracy of 93. **[3]** covers fifteen different classical machine learning models to predict WQI, the most effective being a MLP and a Gradient Boosting model. **[4]** employs time-series models to forecast pH, which is different from our research problem. However, these time-series models obtain notable results, which implies a relationship between past pH values and new pH values. It is also trained on Virginia USGS water datasets, which in turn have been validated by the success of the models.

3. Materials and Methods

This section outlines the methodology used in analyzing and predicting pH.

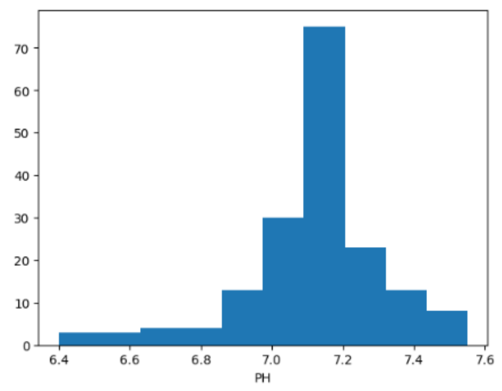
3.1. Dataset

The dataset we are using is an assortment of various climate, landscape, and water quality metrics measured across twenty different watersheds in Fairfax County, Virginia. Because of the proximity of these watersheds to the Chesapeake, we will use them as a proper substitution. This data was taken from the United States Geological Survey. Each metric has an annual recording from 2007-2018. These annual metrics are based on water years, which begin on October 1st and end on September 30th. Metrics are separated into eleven different categories: Water Quality, Climate, Land Use and Land Cover, Housing Units, Wastewater Infrastructure, Stormwater Infrastructure, Stormwater Management Practices, Nitrogen Input, Phosphorus Input, Physical Watershed Setting, and Phosphorus Geochemistry. In total, the raw data set contained ninety-nine metrics, with two-hundred-and-fifty-two samples for each metric. The distribution of the raw dataset is displayed in **Table 1**.

Table 1 Exploratory Data Analysis

Count	23046
Mean	582
Std	2375
Min	-17
25%	1
50%	12
75%	62
Max	49157

Fig. 1 Distribution of pH



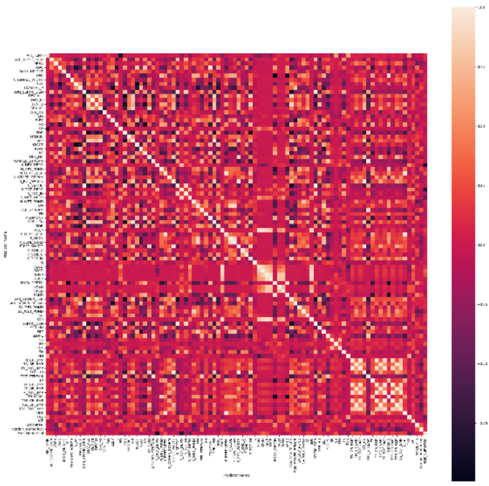
3.2. Data Preprocessing

Before utilizing the dataset, we first pivoted it by Station and Year. Then, we removed all metrics with null values greater than one hundred fifty, and all rows with at least one null value. All Water Quality metrics were measured starting in 2009, so approximately two years of data was removed per station, leaving our dataset only with values from 2009-2018. We then performed discretization to parse Station and Year to be included as features. The preprocessed dataset contains ninety-six unique metrics and one hundred and seventy-six samples for each metric. We then created a separate dataset based on the preprocessed dataset which also included pH lag variables—variables based on past values of pH time series (e.g., prior years). The lag variable dataset only includes data from years 2010-2018, due to the necessity of removing null values associated with the shift in the lag variable column.

3.3. Correlation Analysis

To first get a sense of what features would be best in incorporating into our model, we ran a correlation heatmap (**Fig. 2**) to determine linear correlations between all our metrics. We then narrowed this down to correlations between all our features (used interchangeably with independent variables and input variables) and our intended output: PH. We then filtered out useless features by taking the magnitude of each correlation and removing the corresponding feature if it was below a threshold of 0.1.

Fig. 2 Correlation Heatmap



From the features that were left, the top ten in highest magnitude in order of greatest least were CEC, TRIASSIC, SC, ROCK_DEPTH, SAND, N_DRY_POND, COASTAL_P, WTDEPTH, and SAN_SEWER_REHAB. These features were then used later as a set of inputs for our models. It is worth noting that within the highest correlation variables, most were highly correlated with each other, hinting at levels of multicollinearity that will skew results as shown later when implementing models. All features are normalized through z-score, where μ is the mean, σ is the standard deviation, and *x* is the measured value. Z-score can be calculated as follows:

$$\boldsymbol{Z - score = x - \mu \square \sigma \square}$$

3.4. Prediction of pH Through Machine Learning

Machine Learning is a subset of computer science and statistics that revolves around computer learning algorithms. Training data is passed to these learning algorithms, which they then utilize to build a model. These models can then be used to predict values they were not explicitly told to do so. A model's accuracy is measured by how well it performs in predicting the dependent variable in testing data.

3.4.1. Splitting Data

We tried four models with each feature subset to try and determine which model had the best performance for predicting pH, which feature subset had the best results for predicting PH, and which the best performing combination of model and feature subset. To train and test our models, we used a 4/1 split, where 80% of the data is used as training data, and 20% is used as testing data (**Fig. 3**). Both testing and training data were stratified by Station in order to take account specificities unique to each station (e.g. location and other geological factors).

Fig. 3 Split Visualization

Available Data
Training
Testing
(holdout sample)

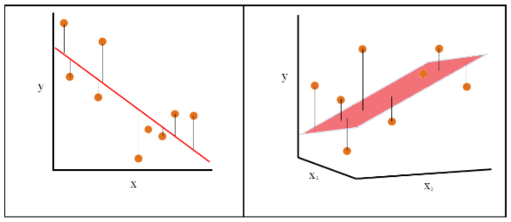
3.4.2. Linear Regression Model

Linear Regression is a supervised learning algorithm that is used to predict continuous (numerical) values given a set of inputs. While training the model, the LR algorithm calculates a cost function (Root Mean Squared Error) to evaluate the accuracy of each predicted value in relation to the true value. We can calculate RMSE using the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

This cost function is then minimized through an optimization algorithm known as gradient descent, which finds the local minimum of a given differentiable function. In our case, this is the RMSE. After this process, it assigns each feature a weight and bias term, establishing a linear relationship between inputs and outputs, which it then uses to predict unknown values given a set of inputs. This linear relationship can be envisioned in a variety of ways, depending on the number of features (**Fig. 4**).

Fig. 4 Visualization of Linear Regression

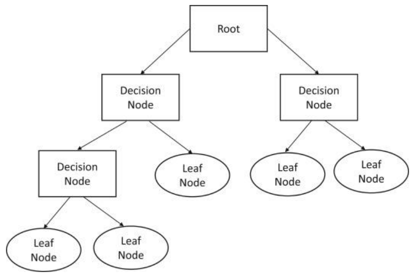


For one feature, it can be envisioned as a line of best fit in R2. For two features, it can be envisioned as a plane of best fit in R3. For everything above two features, it is best interpreted as an m-by-n matrix where n is 1 (row matrix) and m is the number of features, in some vector space. In determining a good use case for a Linear Regression model, several factors must be considered. Key amongst these are linearity, misspecification, and multicollinearity. In order to use a Linear Regression model, data must follow these three constraints. Since there is a linear correlation between inputs and outputs as shown through the correlation analysis, it fits the constraint of linearity. Because we preprocessed the data and removed all null values, we fit the constraint of misspecification. However, since we have many features that are correlated with each other as well as the dependent variable, we do not fit the constraint of multicollinearity. Because multicollinearity generates high variance in the coefficients, it reduces the accuracy of these coefficients, and they become sensitive to small changes in the model. This exacerbates our existing issue of a relatively small dataset, which makes it hard for our model to generalize because of its specificity. However, to test out these assumptions, this model will be evaluated.

3.4.3. Decision Tree Model

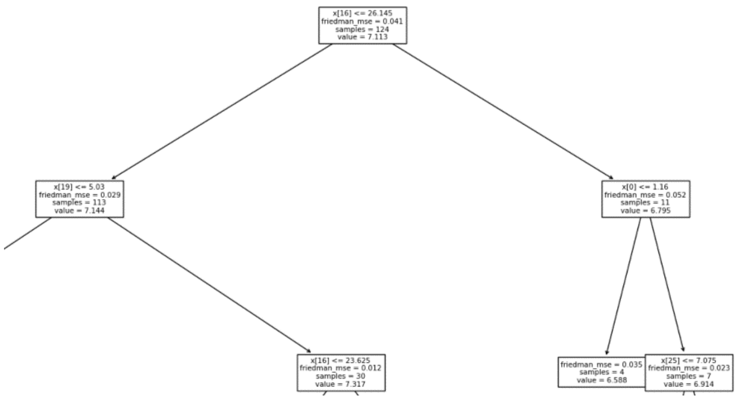
A Decision Tree Regressor, like Linear Regression, is a supervised learning algorithm that is used to predict continuous data. The architecture of a Decision Tree consists of three types of nodes (points of data connected to each other): root nodes, decision nodes, and leaf nodes (**Fig. 5**).

Fig. 5 Node Visualization



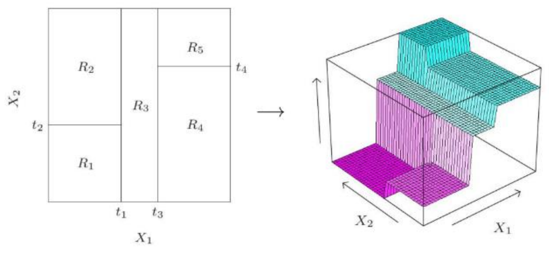
These three types of nodes form a tree structure, where the beginning root node and decision nodes make splits in the data to make predictions (**Fig. 6**). The quality of these splits can be specified when tuning the model, but the standard criterion for determining the quality of splits is MSE or mean squared error. The tree will determine which splits to make by minimizing the MSE of residuals (differences between the actual and predicted values). Once all appropriate splits have been made, the average of all values contained in each grid (if in R2 and there is one feature) formed by the splits will be taken, and these values will be the predicted values for the output or independent variable. As stated, if there is a single feature, the splits of a Decision Tree can be visualized as lines creating grids on the xy-plane. If there are two features, splits can be visualized as planes cutting through R3 to create cubes (**Fig. 7**).

Fig 6. Tree Visualization



For anything above two features, splits can be best interpreted as an m-by-n matrix where m is the number of features and n is the number of splits in some vector space. A downside to trees is that they are prone to overfitting. Overfitting is where a model obtains an extremely high accuracy when training because it memorizes the tiny nuances in the training data. It then performs poorly during testing because it is bad at generalizing given inputs and is fixed to the specificities of the training data. Decision Trees usually overfit because there is not a predetermined max depth that a tree can have, meaning it can create unlimited splits to completely memorize the training data. Fortunately, however, Decision Trees can be tuned to prevent overfitting, which will be discussed later. Despite the drawback of overfitting, Decision Trees are still widely used due to their ability to make predictions through splits, which makes them extremely useful in cases where there isn't a linear relationship between features and a given output. Because of this, tree architectures are commonly used when it comes to quadratic, cubic, quartic, or other complex relationships. In the case of our dataset, although there is some linear correlation between our features and our output, the highest magnitude of correlation is relatively low at around 0.5. Because of this, we will also be using a Decision Tree regressor model in the case of likely non-linear complexities in the data.

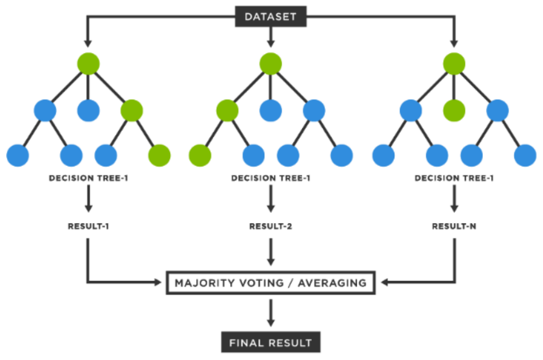
Fig. 7 Split Visualization



3.4.4. Random Forest Model

As the name suggests, a Random Forest Regressor is a supervised learning model that consists of several Decision Trees that are used in conjunction with each other to predict continuous data. First, the algorithm builds new datasets from the data that is passed to it. It does this by first selecting random rows from the original dataset for each new dataset. This process is known as bootstrapping. The new datasets will have the same number of rows as the original dataset. Then, it randomly selects features for each of the new datasets (each dataset will have the same number of features). Finally, it creates the new datasets by taking each randomly selected set of features and filling out an empty dataset with the values of each feature within each randomly selected row. Once these new datasets are completed, the algorithm will then train a tree on each new dataset. Once a set of features are passed to the model, each tree will run the input, and the predicted values will be the average of all results. Combining results from multiple models is known as aggregation (**Fig. 8**).

Fig. 8 Forest Aggregation Visualization



Bootstrapping combined with aggregation is referred to as bagging and is the basis of the Random Forest algorithm. The main benefit of Random Forest is its robustness when it comes to overfitting. The process of bagging ensures insensitivity to training data, which helps greatly in reducing overfitting. Since our data may contain a variety of non-linear relationships but Decision Trees may overfit, we will also be evaluating a Random Forest model.

3.4.5. Gradient Boosted Tree Model

Like Random Forest, Gradient Boosted Trees are a type of ensemble model based on trees, meaning that they use multiple trees in order to obtain predictions. However, whereas Random Forest builds unique trees and combines them in parallel, Gradient Boosted Trees employ a method known as boosting. Boosting combines trees in a sequence such that each new tree corrects the error of the previous tree based on a predetermined loss function (**Fig. 9**). The default value of this loss function, like for Linear Regression, is MSE. Each new tree corrects the error of the previous by first fitting the gradient of the loss function (MSE) with respect to the previous model's output as the dependent variable. Once a new gradient is predicted, the predicted value for the true dependent variable is added with the product of the learning rate (how far to go in the direction of the gradient), and the new gradient (**Fig. 10**). This process repeats itself until the tree decides that suitable predictive values for the true dependent variable have been found. Because there is no predetermined number of times this correction process is executed, Gradient Boosted Trees are prone to overfitting. This, however, like every other model can be tuned accordingly. Gradient Boosted Trees are extremely good at modeling complex relationships between data because of the boosting process that occurs. They are superior in this aspect to even Decision Trees and Random Forest. Because our data may have hidden complex relationships that even Random Forest and Decision Trees cannot model properly, we will be evaluating a Gradient Boosted Tree model.

Fig. 9 Boosting Visualization

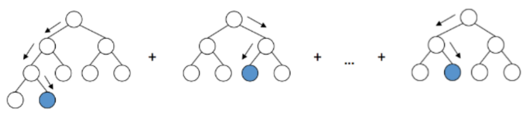
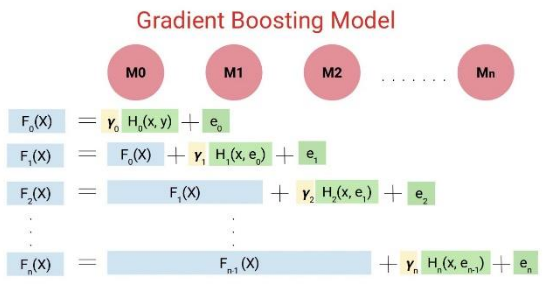


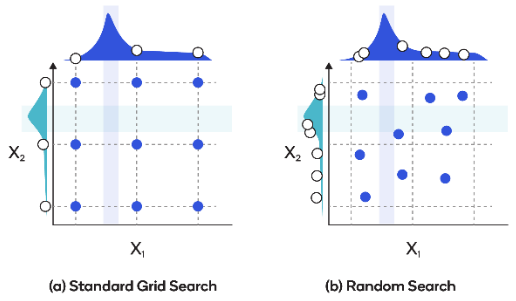
Fig. 10 GBT Equation Visualization


Gradient Boosting Model
M0
M1
M2
Mn
$F_0(X) = \gamma_0 H_0(x, y) + e_0$
$F_1(X) = F_0(X) + \gamma_1 H_1(x, e_0) + e_1$
$F_2(X) = F_1(X) + \gamma_2 H_2(x, e_1) + e_2$
$F_n(X) = F_{n-1}(X) + \gamma_n H_n(x, e_{n-1}) + e_n$

3.4.6. Hyperparameter Tuning

For each model besides Linear Regression, there are certain hyperparameters that can be tuned to increase accuracy and to avoid overfitting. To find optimal hyperparameters for our models, we used a cross validation grid search approach, where we tested different possible combinations of parameters and narrowed down the best ones through a grid of possible values (**Fig. 11**).

Fig. 11 Grid Search Visualization



4. Results and Discussion

This section outlines results from models as well as our interpretation of these results.

4.1 Model Results

Table 2 Best Performing Models

Feature Subset	Best Model	R² Train	R² Test
Water Quality	GBT	0.885	0.560
Climate	RF	0.073	-0.079
Land Use & Land Cover	RF	0.492	**0.672**
Housing Units	RF	0.546	0.595
Wastewater INFSTR	RF	0.693	**0.677**
Stormwater INFSTR	RF	0.621	0.667
Stormwater Management	RF	0.270	0.397
Nitrogen Input	RF	0.872	0.630
Phosphorus Input	RF	0.556	0.562
Physical Watershed SET	RF	0.715	**0.675**
Phosphorus Geochemistry	RF	0.420	0.439
Station	RF	0.510	0.651
Year	RF	0.043	-0.009
Highest Correlation Vars	RF	0.595	0.618
All Variables	RF	0.886	0.635
Lag -1	RF	0.630	0.574

Table 4 Best Models for Combinations of Features

Feature Subsets	Best Model	RMSE Train	R² Train	RMSE Test	R² Test
pH Lag-1 + Physical Watershed + Land Use	DT	0.067	0.887	0.080	0.781
Physical Watershed + pH Lag - 1	DT	0.085	0.822	0.091	0.713
Land Use & Land Cover + pH Lag - 1	DT	0.093	0.785	0.083	0.766
Wastewater INFSTR + pH Lag - 1	DT	0.089	0.805	0.078	**0.789**

Table 3 Evaluation of all Models for Highest Quality Feature Subsets

Feature Subset	LR		DT		RF		GBT	
Land Use and Cover	0.387	0.237	0.570	0.577	0.4923	**0.672**	0.910	0.575
Wastewater Infrastructure	0.303	-0.006	0.829	0.438	0.693	**0.677**	0.872	0.500
Physical Watershed Setting	0.832	0.374	0.821	0.446	0.715	**0.675**	0.819	0.504
pH Lag -1	0.673	0.521	0.636	0.544	0.630	**0.574**	0.748	0.430
Evaluation Metric	R² Train	R² Test	R² Train	R² Test	R² Train	R² Test	R² Train	R² Test

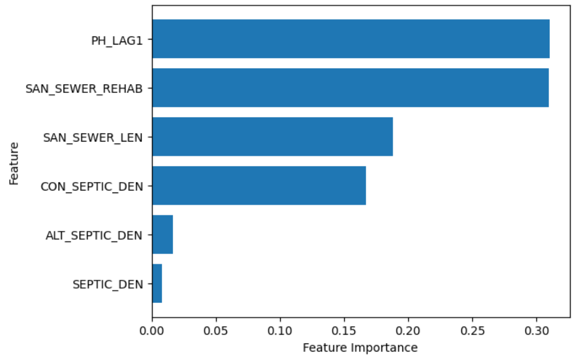
4.2 Discussion

When just training on individual feature subsets, the best performing model by a large majority is Random Forest (**Table 2**). This is likely due to the unique bagging process it executes, which gives it robustness against overfitting and allows it to generalize on the testing data. On average, the Decision Tree models have a higher training accuracy but perform significantly worse than the Random Forest models on testing data. The average training and testing R^2 values for the Decision Tree models was 0.58 and 0.34, respectively. For the Random Forest models, the R^2 values were 0.53 and 0.51. This heavily implies overfitting on the part of the Decision Tree models, which is protected against in the Random Forest models. The only feature subset that had better results for a model other than Random Forest is Water Quality. This can be narrowed down to the fact that Water Quality metrics and pH have an extremely complex relationship that can only be modeled by a Gradient Boosted Tree because of its ability to pick up intricacies in the data. As for the Linear Regression models, multicollinearity likely came into play, as the R^2 for the Linear Regression model trained on the highest correlation features subset has an extremely poor value of -0.91. This multicollinearity most likely also manifests in other feature subsets as well, as features within each feature subset are usually highly correlated with one another. This would help explain the poor performance of the Linear Regression models compared to the others.

Feature subsets that performed the best in **Table 2** were deemed high quality and were evaluated across each type of model as shown in **Table 3.** The pH lag variable is shown in **Table 3.** All feature subsets perform relatively well under every model, which again reinforces their status as high-quality features. This implies a high feature importance in determining PH.

The lag-1 feature doesn't do well when trained on its own, but it boosts R^2 values significantly in conjunction with other data, specifically the highest quality subsets in the data (**Table 4**). Combining the lag-1 feature with the other high-quality subsets gave us our most accurate R^2 value of 0.789. This could be due to some hidden relationship between previous pH values and the other indicators that allow the model to make more accurate predictions. Displaying the feature importance for our best performing models shows us that the lag-1 feature and the percentage of rehabilitated sanitary sewer lines are the two most important indicators in predicting pH (**Fig. 12**).

Fig. 12 Displaying Feature Importance for Best Performing Model



Another interesting thing to note is that Decision Tree models outperform Random Forest models for every combination of lag variable + high-quality subset. Again, this may be due to a hidden association between previous pH values and other indicators. It is most likely too complex for Random Forest models to predict correctly, due to their robustness against overfitting. However, this relationship is likely not as complex as the relationship between Water Quality metrics, as Gradient Boosted Tree models still perform very poorly with even with combinations.

5. Conclusions and Future work

From our model results, the feature subsets that play the biggest role in pH are Land Use and Cover, Wastewater Infrastructure, and Physical Watershed Setting. Our best performing model was a combination of Wastewater Infrastructure and our Lag-1 feature: the pH values of the year prior. Within our best performing model, the individual features deemed most important were our Lag-1 feature and the percentage of rehabilitated sanitary sewer lines. The implications of these results are several. First off, we can clearly establish a relationship between the percentage of rehabilitated sanitary sewer lines, and pH. Knowing this now, it would be possible to conduct further research to deduce exactly what this relationship is. If that relationship is understood, it can be used to regulate pH levels at if they ever fall bellow or above normal thresholds. The same can be said of all other feature subsets as well. We can also establish a direct relationship between pH values of previous years and current pH levels, which supports the conclusions established in [**4**] with the time series models.

As for our models themselves, although they performed decently well, they would need to be trained on larger sets of data and tuned accordingly for higher accuracy values. They are a far way off from replacing sensors and being implemented, but with some extra work and training, it is certainly within the realm of reason.

6. References

[1] Hassan, Md.Mehedi & Hassan, Md & Akter, Laboni & Rahman, Md. Mushfiqur & Zaman, Sadika & Hasib, Khan & Jahan, Nusrat & Smrity, Raisun & Farhana, Jerin & Raihan, M. & Mollick, Swarnali. (2021). Efficient Prediction of Water Quality Index (WQI) Using Machine Learning Algorithms. 1. 1-12. 10.2991/hcis.k.211203.001.

[2] U. Shafi, R. Mumtaz, H. Anwar, A. M. Qamar and H. Khurshid, "Surface Water Pollution Detection using Internet of Things," *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*, Islamabad, Pakistan, 2018, pp. 92-96, doi: 10.1109/HONET.2018.8551341.

[3] Ahmed, Umair, Rafia Mumtaz, Hirra Anwar, Asad A. Shah, Rabia Irfan, and José García-Nieto. 2019. "Efficient Water Quality Prediction Using Supervised Machine Learning" *Water* 11, no. 11: 2210. https://doi.org/10.3390/w11112210

[4] Srivastava, A., Cano, A. Analysis and forecasting of rivers pH level using deep learning. *Prog Artif Intell* **11**, 181–191 (2022). https://doi.org/10.1007/s13748-021-00270-2