Water Potability Prediction using Random Forest, Support Vector Machine, and K-Nearest Neighbor

Palaniappan Balasubramanian - 220196730

***Abstract*— This project predicts the potability of the water using the concentrations of sulphate, trihalomethanes, and chloramines and the properties of water such as turbidity, solids, organic carbon, hardness, pH, and conductivity. The comparison of three Machine Learning methodologies .i.e. Random Forest, Support Vector Machine and K-Nearest Neighbor is carried out. This is a classification problem classifying potable water to be 1 and non-potable water to be 0.**

I. INTRODUCTION

All life on Earth depends on water, which is one of the planet's most important resources. Water potability is a vital issue since not all water is safe to drink. Water that is fit for human consumption and devoid of dangerous impurities is referred to as potable water. The project will help find the suitable water for drinking.

Harmful pollutants are one of the biggest issues with water potability. These pollutants can originate from a variety of places, including sewage, agricultural runoff, and industrial waste. Bacteria, viruses, parasites, heavy metals, pesticides, and chemicals are a few of the frequent pollutants found in water sources. Without taking into account the relationships between the chemical, biological, and physical variables, M. A. Tirabassi et al. devised a statistical model to predict the water quality [1]. Based on Kalman-filtering and self-adaptive approaches, H. C. Guo et al. suggested a stochastic water-quality prediction system that was created to expose the hazard characteristics of various parameters. The BOD and DO concentrations in the Yilou River were predicted using the algorithm [2].

The difficulty in identifying contaminants is another problem with water potability. Water can include dangerous elements like lead and arsenic without giving off any overt symptoms or flavors. This can make it difficult to locate sources of polluted water, particularly in places where water testing is not easily accessible. A appropriate categorization model based on machine learning methods was proposed by Salisu Yusuf Muhammad et al. [4]. Five classification algorithms—Naive Bayes, K star, Bagging, J48, and Conjunctive rule—have been compared in order to identify the crucial elements that contributed to categorising the water quality of the Kinta River in Perak, Malaysia.

So, I was inspired to utilize this Water Quality dataset to analyze what constitutes safe, potable water and to apply machine learning to it so that it could discriminate between potable and non-potable water.

II. BACKGROUND

The dataset is based on water potability. This dataset considers certain features present in water that helps us distinguish whether the water is drinkable or not. Consuming pure water is most important to avoid getting affected. The features of the dataset include:

1. ph: Water's pH (0 to 14).
2. Hardness: The amount of soap that may be precipitated per litre of water.
3. Solids: The total amount of dissolved solids in ppm.
4. Chloramines: Chloramine concentration in ppm.
5. Sulfate: The amount of dissolved sulfates expressed in mg/L.
6. Water's electrical conductivity: measured in S/cm.
7. Organic carbon: The percentage of organic carbon.
8. Trihalomethanes: Trihalomethane concentration in g/L.
9. Turbidity: The amount of light that water emits as measured in NTU.
10. Potability: A measure of a liquid's suitability for human ingestion. Potable is one and unpotable is zero.

III. ANALYZING THE DATA

- The data has 10 variables and 3276 data points.
- The describe() function is used to find statistical calculations from the dataset such as count. Mean. Mode, median, standard deviation, etc.
- We would obviously prefer a model that would have more false negatives rather than a model that has more false positives. The more significant label ("Not potable") is the one with more samples. The non-potable feature constitutes 61% and potable features constitute 39% of the dataset.

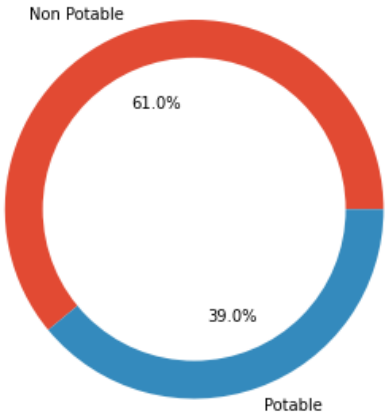

Fig. 1. Pie chart of the Potability Feature

- Since our output label is binary and our features are continuous, it appears that there is no linear or ranked

correlation between them. As a result, standard linear correlation coefficients don't accurately reflect the links between our features and the target variable. We will do additional in-depth research later in this coursework in an effort to unearth some of the correlations buried in our data [3,6].

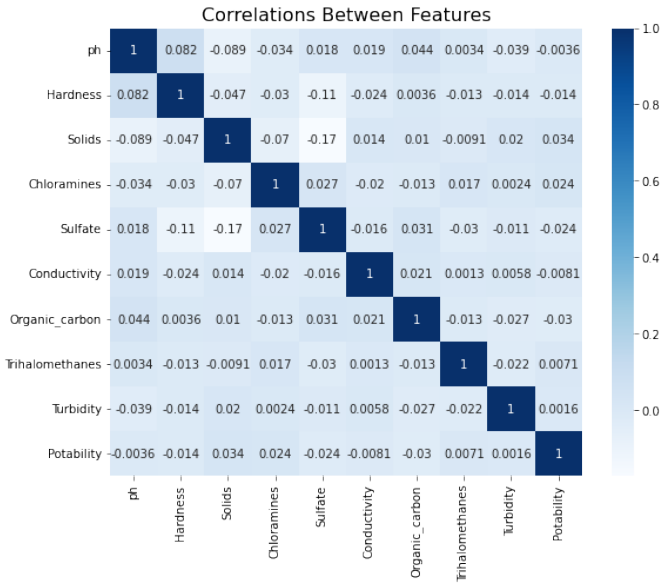

Fig. 2. Correlation between the variables

- When we divide the distribution of all of our features by the target label, we can see that some of them differ from one another. This is an important finding that may guide our decision over which characteristics to use to train our models. A more thorough investigation is needed to support whatever theory we may have at this time based just on looking at the distribution plots, in order to better comprehend the variations between the characteristics with regard to the target label.

IV. Data Processing

A. Missing Null Values

The missingno library is used to fill the missing or null values. The features such as pH, Sulfate, and Trihalomethanes has null values. To fill the missing values, the mean of the whole column is taken and filled values in place of null values. Figure 3 shows the null values present in the dataset. It was found that the null values were in pH (491 values), Sulfate (781 values) and Trihalomethanes (162 values). The Figure 4 shows the null values being filled after using this library.

B. Normalization of Data

Minmax scaler library is used to normalize the data between 0 to 1. This library is mainly used because it normalizes the data and at the same time holds the original shape of the dataset. After normalization, an example of a 2000th data is given below.

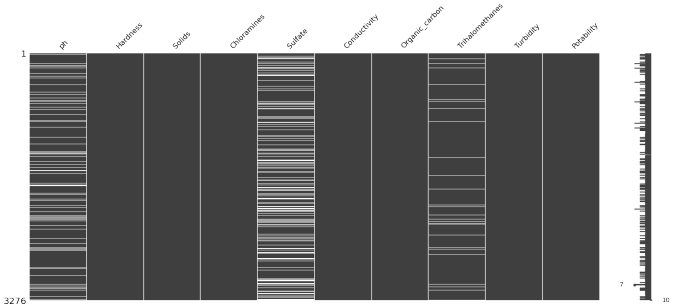

Fig. 3. The null values in the dataset

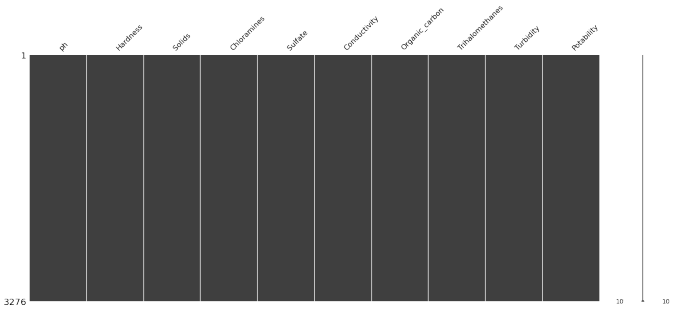

Fig. 4. After using the missingno library

array([0.53157308, 0.5344075 , 0.56225358, 0.63728749, 0.47108461, 0.29483037, 0.64222276, 0.64269865, 0.27314259])

C. Data Skewing

Examining the correlation between the numbers with the help of the pandas skew function. According to the skewness value, values between -0.5 and 0.5 will be regarded as having a normal distribution; otherwise, they will be skewed. The table represents the skewness of each feature in the dataset.

Features	Skewness value
Solids	0.621634
Potability	0.450784
Conductivity	0.264490
ph	0.027796
Organic carbon	0.025533
Turbidity	-0.007817
chloramines	-0.012098
Hardness	-0.039342
Sulfate	-0.041184
Trihalomethanes	-0.085161

D. Balancing the dataset

The input are the variables except the potability column. The output to be predicted is the final column (potability) either "1" or "0". The train_test_split from sklearn.model_selection is used to perform splitting the dataset into training and testing dataset. The training set is

75% of the dataset and the testing set is the remaining 25% of the dataset.

For balancing the data, I am using SMOTE. SMOTE (Synthetic Minority Over-sampling Method) is a well-known machine learning approach that addresses the problem of unbalanced datasets. Imbalanced datasets make it harder for the algorithm to effectively learn from and predict from the data since the quantity of samples in one class is much smaller than the other. In order to balance the dataset and enhance the algorithm's performance, SMOTE uses generated samples of the minority class. [3,7] By interpolating between minority class examples that already exist, the method develops synthetic samples that lie between the existing samples. The program then adds a few of these artificial examples to the training set at random, helping to balance the dataset. The data were initially 1496 data points for non-potable and 961 data points for potable. After balancing the data with SMOTE, each class gets equal data points. The output before and after applying SMOTE is given below.

Oversampling - Balancing the data by SMOTE
Before: Counter(0: 1496, 1: 961)
After: Counter(0: 1496, 1: 1496)

E. Standardization of Data

Standard scaler library from sklearn.preprocessing is used to standsrdize the data. This library transforms the data with the help of mean 0 and standard deviation 1. It maintains the useful information of the outliers. This estimator fits the training dataset and transforms both training and test dataset using the same estimator.The Figure 5 below consists of two figures, first one is the data before standardization and the other figure is the data after standardization.

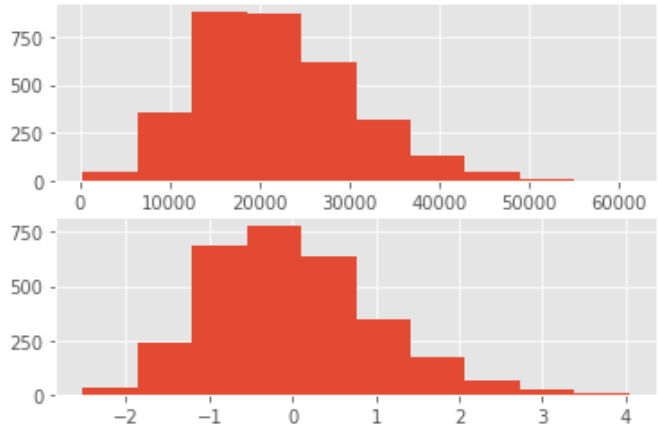

Fig. 5. Standardization of the dataset

V. Methodologies

Unsupervised learning techniques are used for this dataset as it is a classification problem. There are three ML techniques that predict the potability of the water with greater accuracy. The three ML models are Random Forest, Support Vector Machine, and K-Nearest Neighbor. These algorithms are performed using the sckit-learn library. This section provides an overview of how each classification technique trains the dataset.

A. Random Forest

A branch of the Decision Tree classifier is the Random Forests classifier. It is made up of several unique decision trees. The class that obtains the most votes is chosen as the final class given to the object when different decision trees yield various class values for the same object. The crucial aspect is how strongly connected these separate decision trees are with one another. The random forest is hence quite potent. An inaccuracy introduced to one tree may not have any impact on the other since the correlation between the two trees is new. These uncorrelated models use randomness to provide the right class value for the object. Feature randomness and bagging are two methods for assuring a high degree of randomness [9].

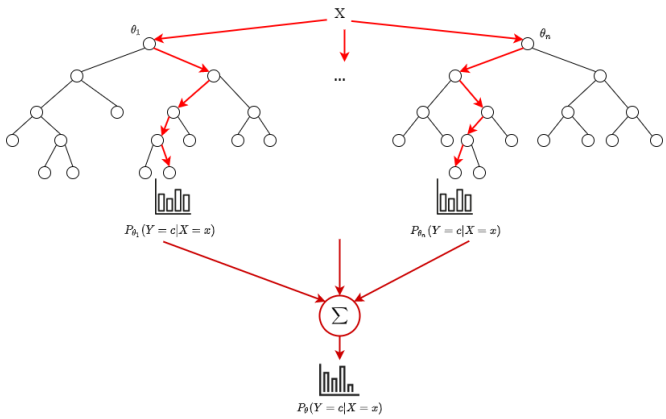

Fig. 6. Classification of Random Forest Classifier

B. Support Vector Machine (SVM)

A popular supervised machine learning approach for classification issues is support vector machines (SVM). The algorithm is built around a kernel technique that extracts crucial information from the supplied data. The algorithm's objective is to find the best border and make an effort to demarcate the data using the labels. Where the new data belongs will be determined with the aid of this necessary delineation [5,8].

C. K-Nearest Neighbor (KNN)

The 1951 invention of KNN by Evelyn Fix and Joseph Hodges relies on the notion that data belonging to the same class coexist. Proximity, distance, and closeness are the three fundamental concepts that make up the K Neighbors classifier. It employs a metric parameter that establishes the formula for calculating this closeness [10]. The Minkowski distance is then used. It can be described using the following formula:

$$\left(\sum_{i=1}^{n}(|x_i = y_i|)^p\right)^{\frac{1}{p}} \quad (1)$$

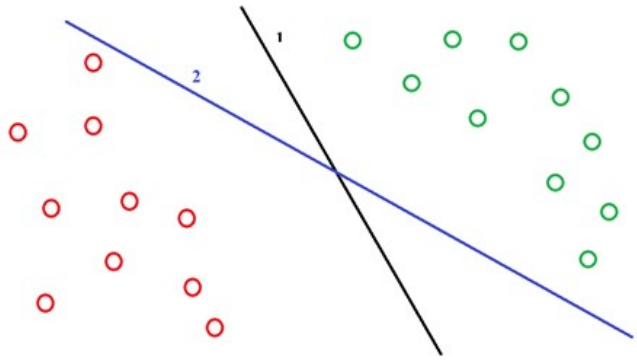

Fig. 7. Classification of SVM

It assigns a class to an object by considering the class of its K (which is a small positive integer) nearest objects' classes.

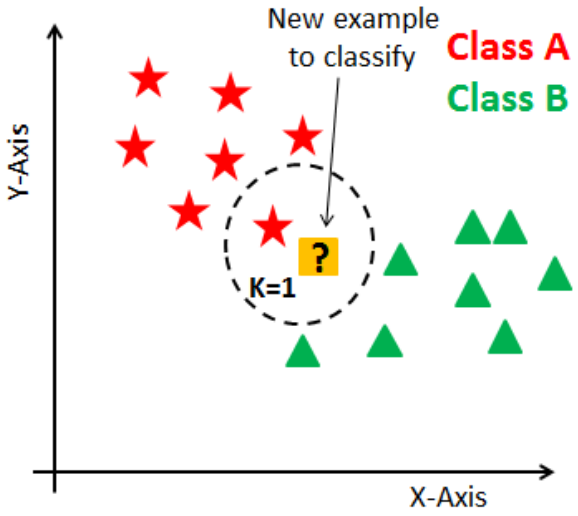

Fig. 8. Classification of KNN

VI. Analysis, Testing, and Results

A. Applying Cross Validation to find the best-performing models and their hyperparameters

The Cross Validation to find the best ML model and parameters that helps the most in determining high accuracy. The most commonly used methods are GridSearchCV and RandomizedSearchCV. GridsearchCV is used to address this problem. The performance of a model can be significantly affected by hyperparameters, which are parameters that are established before the model is trained. [3] GridSearchCV operates by methodically experimenting with various combinations of hyperparameters and assessing the model's performance with each combination. The most effective hyperparameters for a particular dataset and model may be found using this effective approach, although it can be computationally costly. Despite this, GridSearchCV is frequently employed in the machine learning community since it significantly enhances model performance. The table below shows the significant findings after applying this cross validation technique.

ML model	Best Score	Best Parameters
Decision Tree	0.59	'max_depth': 6
Support Vector Machine	0.69	'C': 10, 'kernel': 'rbf'
Random Forest	0.71	'n_estimators': 20
Logistic Regression	0.52	'C': 1.0, 'solver': 'newton-cg'
K-Nearest Neighbour	0.74	'n_neighbors': 1

B. Comparing the Best Three ML Models

The dataset is split into 75% training set and 25% testing set. According to the cross validation results, the three best ML models were Random forest, Support Vector Machine, and K-Nearest Neighbor.The Random forest algorithm performed the best with the parameter "n_estimator=20" , the Support vector machine algorithm performed the best with the parameter "C=10 and kernel='rbf'", and K-nearest neighbor performed the best with the parameter "n_neighbors=1". The Figure 9 shows the comparison of the accuracy of the ML models. In which the SVM algorithm performs the best with accuracy 66%. Eventhough, other classifier's accuracy doesnt vary much .i.e. Random forest with 64% and KNN with 59%.

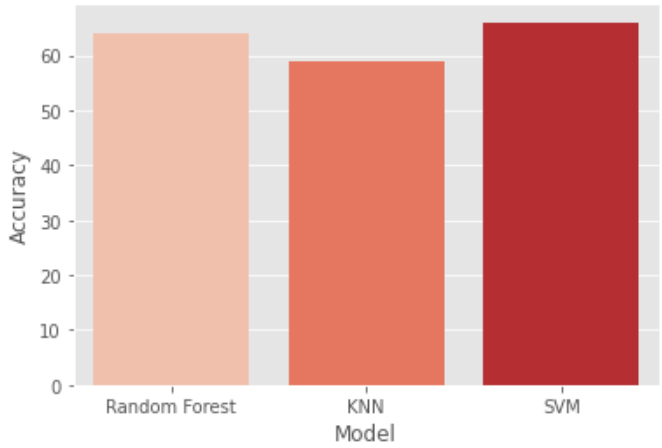

Fig. 9. Accuracy comparison of the ML Models

The Figure 10, Figure 11, and Figure 12 are the confusion matrices of the ML models on the test data. From the confusion matrices irrespective of which model, it is predicting the non-potable water to be '0' well and the model is struggling to predict the potable water to be '1'.

The table below shows the training accuracy, test accuracy, precision, recall, and AUC of the ML models. The training accuracy is relatively high compared to the test accuracy for the models. So we can tell that the models are struggling when coming to the test data.

ML model used	Train Accuracy	Test Accuracy	Precision	Recall	AUC
Random Forest	1.0000	0.6532	0.565737	0.447950	0.615409
Support Vector Machine	0.7550	0.6459	0.546713	0.498423	0.618733
K-Nearest Neighbour	0.7988	0.5897	0.474394	0.555205	0.583379

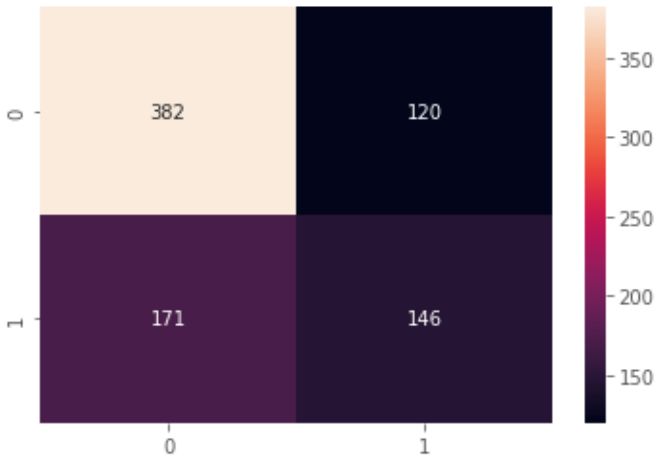

Fig. 10. Confusion Matrix for Random Forest

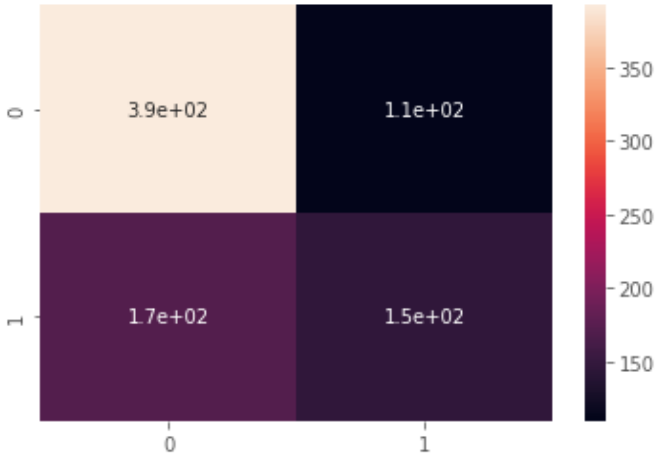

Fig. 11. Confusion Matrix for SVM

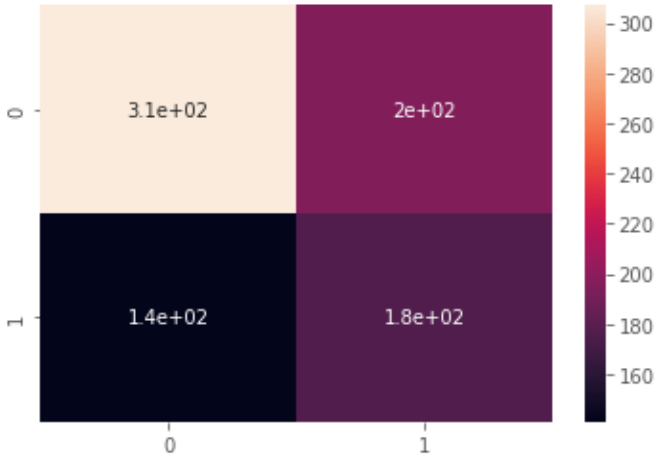

Fig. 12. Confusion Matrix for KNN

The Figure 13 shows the ROC curve and the corresponding AUC value for each ML model. The ROC (Receiver Operator Characteristic) curve shows the performance rate of the model at all classification limits. It is plotted using the true positive rate and the false positive rate. The AUC (Area Under Curve), as the name suggests, it is the area underneath the ROC curve. It shows the predictability of the Machine Learning models used. Higher the AUC value, the higher the predictability. For the ML model used in this project, the AUC values were 61% for the Random forest algorithm, 62% for the Support Vector Machine algorithm, and 58% for the K-Nearest Neighbour algorithm.

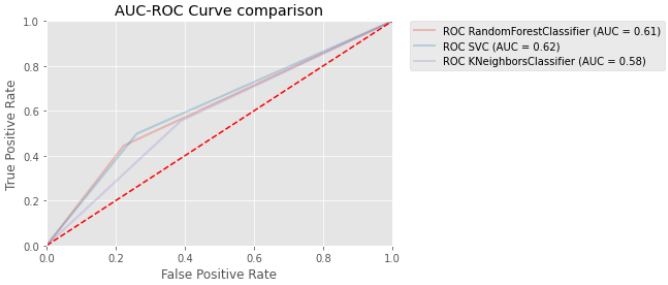

Fig. 13. ROC Curve for the ML Models

VII. Conclusion

The project reviewed in this analysis presented several approaches for predicting the potability of water using machine learning models. I compared three ML models and found that it classifies the data very well. The results demonstrated that by pre-processing the data to fill null values and standardizing and normalizing the features, the ML models could accurately predict the potability of water with an average accuracy of 63%. The ML models perform well when predicting non-potable water and struggles when predicting potable water. All three ML models have predicted AUC scores also to be around 60%.

However, the training accuracy was found to be higher than the test accuracy, indicating that the models were struggling to generalize to new data. To overcome this limitation, future research could involve the use of deep learning methods to gain a better understanding of the dataset and improve the accuracy of the models.

Another limitation of the research was that the models were built using only 9 variables, and there could be several other features that contribute to predicting potable water. Future research could involve the inclusion of additional features to improve the accuracy of the models.

Furthermore, the amount of data used for training and testing the models was limited, with only around 3000 data points. Increasing the dataset size could improve the accuracy of the models further, and future research could involve the collection of more data to achieve this goal.

Overall, the reviewed papers have shown that machine learning models can be useful in predicting the potability of water. By pre-processing the data and using cross-validation techniques, the models could achieve high accuracy levels. However, there are still limitations to these models, and further research is necessary to improve their accuracy and

effectiveness. With continued research and development, machine learning models could be used to ensure access to safe drinking water in areas where water quality is a concern.

REFERENCES

[1] M. A. Tirabassi, "A statistically based mathematical water quality model for a non-estuarine river system1", JAWRA Journal of the American Water Resources Association, vol. 7, pp. 1221-1237, December 1971.

[2] H. C. Guo, L. Liu and G. H. Huang, "A stochastic water quality forecasting system for the Yiluo River", Journal of Environmental Informatics, vol. 1, no. 2, pp. 18-32, 2003.

[3] Jinal Patel, Charmi Amipara, Tariq Ahamed Ahanger, Komal Ladhva, Rajeev Kumar Gupta, Hashem O. Alsaab, Yusuf S. Althobaiti, Rajnish Ratna, "A Machine Learning-Based Water Potability Prediction Model by Using Synthetic Minority Oversampling Technique and Explainable AI", Computational Intelligence and Neuroscience, vol. 2022, Article ID 9283293, 15 pages, 2022.

[4] Salisu Yusuf Muhammad, Mokhairi Makhtar, Azilawati Rozaimee, Azwa Abdul Aziz and Azrul Amri Jamal, "Classification model for water quality using machine learning techniques", International Journal of software engineering and its applications, vol. 9, no. 6, pp. 45-52, 2015.

[5] A.H. Haghiabi, A.H. Nasrolahi and A. Parsaie, "Water quality prediction using machine learning methods", Water Quality Research Journal, vol. 53, no. 1, pp. 3-13, 2018.

[6] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, "Efficient water quality prediction using supervised machine learning," Water, vol. 11, p. 2210, 2019.

[7] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," Information Sciences, vol. 465, pp. 1–20, 2018.

[8] Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. 2001, 2, 45–66

[9] Liaw, A.; Wiener, M. Classification and regression by random Forest. R News 2002, 2, 18–22.

[10] Beyer, K.; Goldstein, J.; Ramakrishnan, R.; Shaft, U. When is "nearest neighbor" meaningful? In Proceedings of the International Conference on Database Theory, Jerusalem, Israel, 10–12 January 1999; pp. 217–235