Available online at
https://mijuitm.com.my

e-ISSN: 2735-0703

Mathematical Sciences and Informatics Journal 7(1) 2026, 187 – 203.

Mathematical Sciences and Informatics Journal

Comparative Study of Preprocessing Techniques for Real-Time Prediction of Water Quality Index

Muhamad Danish Saiful Riza[1], Fakhitah Ridzuan[1*], Nurzulaikha Abdullah[2], Mohd Hakimi Aiman Ibrahim[1]

[1]*Faculty of Data Science and Computing, Universiti Malaysia Kelantan, P.O.BOX 36, Pengkalan Chepa, 16100 Kota Bharu, KelantanKota Bharu, Kelantan, Malaysia*
[2]*Faculty of Defence Science and Technology, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia*

ARTICLE INFO

Article history:
Received 3 February 2026
Revised 15 March 2026
Accepted 23 April 2026
Online first
Published 30 April 2026

Keywords:
Water quality index
Machine learning
Preprocessing
Regression
Outlier

DOI:
10.24191/mij.v7i1.11950

ABSTRACT

Water quality is a fundamental component in human health, environmental sustainability, and industrial utilization. However, conventional methods in monitoring water quality are slow and cannot operate in real-time. This issue highlights the need for an efficient forecasting approach for water quality assessment. This research aims to explore how a machine learning-based framework can be developed to predict the Water Quality Index (WQI) based on eleven key indicators of water quality. Six regression algorithms were compared in this study which are Support Vector Machine, Multi-Layer Perceptron, Decision Tree, Random Forest, XGBoost, and Gradient Boosting. To ensure model reliability, comprehensive data preprocessing steps were implemented. This involved addressing missing values, outlier elimination, and normalizing data to prevent poor model performance caused by variable imbalance or data distortion. In addition, feature selection and Principal Component Analysis were employed to enhance optimal input features. The models were evaluated using R Squared (R^2), Root Mean Squared Error (RMSE) and training time. Among all models, Gradient Boosting with normalized data preprocessing achieved the highest overall performance with an R^2 value of 0.9943 and an RMSE of just 0.5457. The findings demonstrated that integrating machine learning with effective preprocessing substantially improves the accuracy of WQI prediction. Consequently, this approach can provide a robust foundation for developing reliable and scalable real-time water quality monitoring systems.

[1*] Corresponding author. *E-mail address*: fakhitah.r@umk.edu.my
https://doi.org/10.24191/mij.v7i1.11950

1. INTRODUCTION

Water plays important roles in public health, environmental sustainability, and socio-economic progress. It supports human life, ecosystems and drives economic activities. However, its quality is under severe threat from pollution. The current global landscape has transformed dramatically due to agricultural expansion and urbanization, causing significant losses of forests and wetlands which are critical for water quality (European Commission, 2025). Water Quality Index (WQI) is a numerical tool used to summarize complex water quality data into a single value that reflects the overall quality of water. It typically combines multiple physical, chemical, and biological parameters of water into one standardized score, which usually ranges from 0 to 100 (Chidiac et al., 2023). A higher WQI value indicates better water quality. As an accessible and interpretable indicator, the WQI facilitates the assessment of water suitability for various purposes, including drinking, recreation, irrigation, and aquatic life support. It enables stakeholders, such as environmental managers and policymakers, to classify and compare water bodies on a quality scale, from "excellent" to "poor", thus supporting informed decision-making and effective environmental governance.

Despite its utility, the traditional WQI calculation, which often relies on complex mathematical formulas, fixed weighting schemes, and sporadic manual sampling, can be time-consuming and resource intensive. The process is also sometimes slow to reflect rapidly changing environmental conditions. These limitations highlight a critical need for more efficient, dynamic, and accurate methods for water quality assessment. Machine learning (ML) has emerged as a powerful tool in enhancing the prediction and assessment of the WQI, addressing many limitations of traditional WQI methods. Researchers have widely applied various ML models to analyze complex water quality datasets, identify patterns, and provide more accurate and timely predictions. These ML applications have been successful in enabling real-time prediction and management of water quality, crucial for proactive decision-making and policy formulation (Kim et al., 2024). By leveraging advanced algorithms, ML models can continuously analyze incoming data streams from various monitoring sensors, rapidly detecting changes in water quality parameters that may indicate pollution events or emerging risks. This real-time analytical capability allows environmental agencies and water resource managers to implement timely interventions, thereby preventing potential ecological damage and safeguarding public health. Moreover, ML algorithms excel at processing extensive and intricate water quality datasets that often contain nonlinear relationships, missing values, and multivariate dependencies. This capability enables them to overcome many of the limitations associated with conventional, labor-intensive monitoring and assessment methods, which are typically slower, less adaptive, and dependent on periodic manual sampling (Zhu et al., 2022).

Despite these advancements, several challenges and gaps persist in data preprocessing for ML-based WQI models. A pervasive issue is missing data, frequently encountered in water quality datasets due to equipment malfunctions, network limitations, or human error during sample collection (Sierra-Porta, 2024). The presence of such gaps can introduce bias and significantly compromise the accuracy and reliability of machine-learning models. In the context of automated WQI detection, data preprocessing plays a critical role in ensuring the integrity and performance of predictive models. Preprocessing techniques for automated WQI detection are important because they clean, structure, and refine raw water quality data to ensure it is accurate, concise, and usable for predictive models. Proper preprocessing improves the quality and reliability of the data by handling missing values, reducing noise, and extracting relevant features, which directly enhances the accuracy, stability, and efficiency of water quality prediction and assessment systems.

This emphasis on preprocessing aligns closely with standard practices in ML, where data preparation is considered a foundational step that significantly influences model outcomes. The phase of data preprocessing is a critical step in ML, that hugely impacts the performance and accuracy of models. The common pre-processing techniques of data in ML are data cleaning data scaling, data transformation, and data reduction. Most of these preprocessing techniques enhance the quality of data to be used in AI models

https://doi.org/10.24191/mij.v7i1.11950

and enhance the efficiency of data extraction for accurate and transformed datasets that can be analysed and interpreted from a wide range of domains (Bala & Behal, 2024). Data cleaning is an effective method of improving the quality of datasets and making data research conclusions more accurate. It is a step-by-step process of finding out and correcting including those that may occur in the process of data collection and analysis. Data cleaning is needed to improve model fit paying special attention to noise, missing values and data inconsistencies (Li et al., 2023).

Therefore, the primary objective of this paper is to conduct a comprehensive comparative study of various data preprocessing techniques and their impact on the predictive accuracy and robustness of ML models for WQI prediction. This paper seeks to: (1) evaluate the efficacy of different preprocessing methods on WQI prediction performance; (2) compare the effect of various data scaling and normalization techniques on the stability and accuracy of different ML algorithms; and (3) identify optimal parameters that enhance the reliability of WQI prediction models.

2. METHODOLOGY

2.1 Dataset Description

In this research, the dataset used was published on the 10th of January 2025 by Rahman et al. (2025). The dataset was recorded since 2007 until 2023 in Cork Harbour, Ireland. This dataset contains 7,790 rows of daily water-monitoring records and captures 11 parameters of water quality based on Table 1.

Table 1. Dataset description

Variable Name	Description	Unit
WaterbodyName	Name of the water-body region	—
Years	Year of the sample collection	yyyy
SampleDate	Date of the sample collection	dd-mm-yyyy
Alkalinity-total (as CaCO3)	Alkalinity level in water	mg/l
Ammonia-Total (as N)	Concentration of ammonia	mg/l
BOD - 5 days (Total)	Biochemical oxygen demand over 5 days	mg/l
Chloride	Concentration of chloride	mg/l
Conductivity @25°C	Conductivity at 25°C	µS/cm
Dissolved Oxygen	Concentration of dissolved oxygen	mg/l
ortho-Phosphate (as P) - unspecified	Concentration of orthophosphates	mg/l
pH	pH of the water	pH units
Temperature	Temperature of the sample	°C (Celsius)
Total Hardness (as CaCO3)	Concentration of total hardness	mg/l
True Colour	Colour measurement of the water	TCU
WQI Value	Water Quality Index value (calculated)	Range 0 – 100
Label	Water quality classification based on WQI	—

Each parameter plays an important role in measuring water quality. Alkalinity measures the ability of the water to neutralize acids, and it is an important element for keeping stable pH levels. Ammonia, frequently a result of agricultural runoff, is toxic to aquatic life at high levels. BOD is an indication of the amount of organic material that is present in the water samples, bigger BOD values indicate greater pollution in the water. Chloride levels can indicate the result of industrial waste or saltwater intrusion. Conductivity measures the water's ability to carry electricity, which relates to how much dissolved salt is in the water. Dissolved Oxygen is essential for aquatic organisms, and its depletion causes dead zones. Orthophosphate is an important nutrient which if in excess can also result in algal blooms. The pH level measures the water's acidity and alkalinity condition, and influences both chemical and biological

https://doi.org/10.24191/mij.v7i1.11950

processes. Temperature affects the solubility of gases and the biological activity of aquatic animals. The Total Hardness, which is defined by the concentration of calcium and magnesium, has an influence on the usability of water. Finally, True color can be a sign of the dissolved organic material or pollutants. In combination, these parameters offer a comprehensive picture of water quality, allowing researchers to be in better position to forecast changes in the water and to better manage drinking-water supplies.

2.2 Data Preprocessing

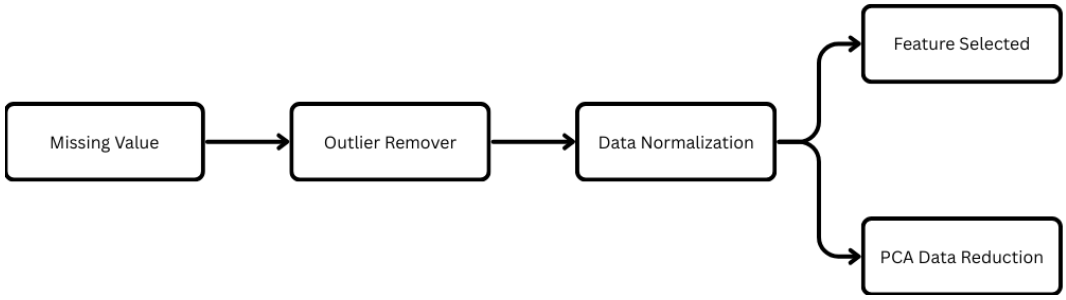

Fig 1. Flow of Data Preprocessing

The first crucial step before model training involves checking for complete data in the dataset. Data failures and null values within datasets introduce biases that lead to inaccurate predictions, undermining the validity of the analysis. Based on Fig. 1, these are the preprocessing steps which included three fundamental actions that were used in this study. First, handling missing values, then removing outliers through IQR methods, and lastly applying Standard Scalar normalization to achieve consistent feature ranges. However, based on the initial screening, there were no null values present in the dataset. Next, outlier analysis was conducted. Outliers were identified systematically using the interquartile range (IQR) procedure for each feature. IQR is the difference between the third and first quartile of a variable's distribution. Outliers of the observations were defined to be out of the range of the following:

$$\text{Lower Bound} = Q1 - 1.5 \times IQR \quad (1)$$

$$\text{Upper Bound} = Q3 + 1.5 \times IQR \quad (2)$$

The values which were less than the lower bound or greater than the upper bound were considered as outliers and then removed (Montgomery & Runger, 2011). The presence of outliers in datasets creates major distortions of statistical tests which lead to training biases that affect ML model performance negatively (Yasnitsky & Plotnikova, 2024). An outlier detection and removal process served as a pre-development step to reduce the risks. Fig.2 illustrates the original data prior to cleaning, highlighting the presence of extreme values across multiple parameters. Specifically, pronounced right-skewness and extreme observations were observed in variables such as Ammonia, BOD, Chloride, Conductivity, and True Color. Stable parameters like pH and Temperature showed noticeable numbers of outliers according to the results. The distribution of all features improved extensively after the IQR method implementation. The application of the IQR method reduced skewness levels while compacting value ranges which eliminated outlier occurrences. The cleaning phase remains essential because it enables upcoming ML models to learn from unbiased data representations which improves the predictive and overall generalization capabilities.

To improve the performance of the model evaluation, feature scaling was conducted since it was necessary to work with datasets that have variables with varying scale ranges. Feature scaling is an important data preprocessing step in ML that involves transforming the numerical independent variables of

https://doi.org/10.24191/mij.v7i1.11950

a dataset to a common scale or range (Kang & Tian, 2018). This is done to prevent features with larger magnitudes from dominating model training. Common scaling methods include Normalization and Standardization. Normalization scales data to a specific range while keeping the differences between values. This step is crucial to algorithms that depend on the size of the data for computations to prevent producing distorted or incorrect outcomes (Dhawas et al., 2023). Additionally, data enrichment increases the quality of data by incorporating related and useful information that increases the value of analytics derived from data (Jansen et al., 2023). On the other hand, standardization transforms data to have a mean of 0 and a standard deviation of 1. It ensures all features contribute equally to the model and accelerates the convergence of optimization algorithms like gradient descent (Ozsahin et al., 2022).

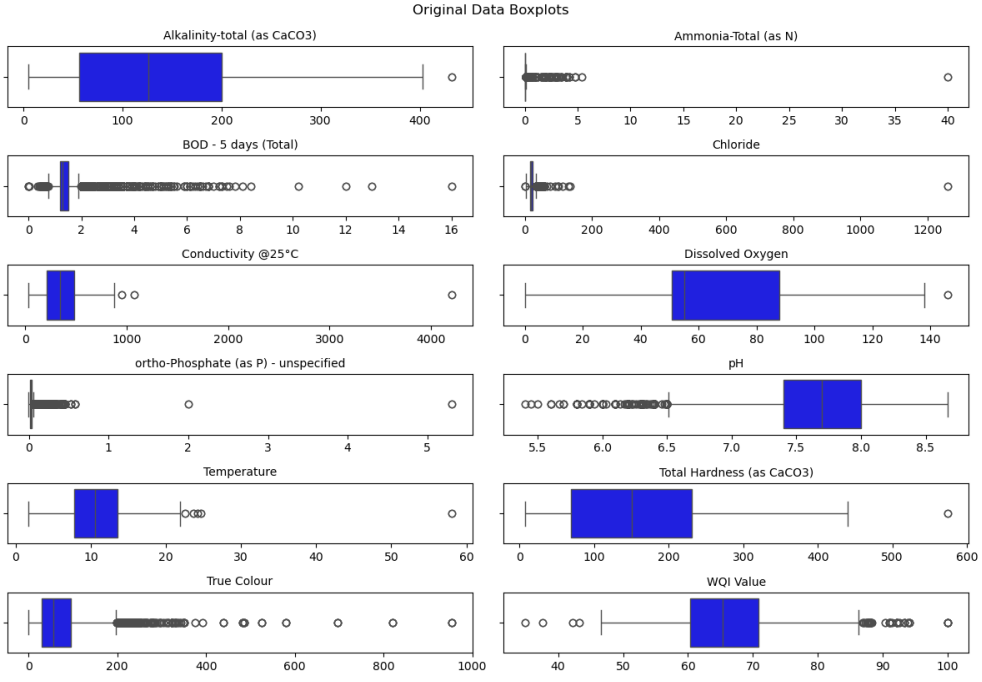

Fig. 2. Original Distribution of the Data

To improve the performance of the model evaluation, feature scaling was conducted since it was necessary to work with datasets that have variables with varying scale ranges as shown on Fig. 2. Feature scaling is an important data preprocessing step in ML that involves transforming the numerical independent variables of a dataset to a common scale or range (Kang & Tian, 2018). This is done to prevent features with larger magnitudes from dominating model training. Common scaling methods include Normalization and Standardization. Normalization scales data to a specific range while keeping the differences between values. This step is crucial to algorithms that depend on the size of the data for computations to prevent producing distorted or incorrect outcomes (Dhawas et al., 2023). Additionally, data enrichment increases the quality of data by incorporating related and useful information that increases the value of analytics derived from data (Jansen et al., 2023). On the other hand, standardization transforms data to have a mean of 0 and a standard deviation of 1. It ensures all features contribute equally to the model and accelerates the convergence of optimization algorithms like gradient descent (Ozsahin et al., 2022).

The performance of some ML models will be affected by inconsistent feature scales because these classifiers heavily depend on the magnitude values of their inputs. Improper scaling leads models to

https://doi.org/10.24191/mij.v7i1.11950

demonstrate slower convergence together with unstable optimization during training as well as biased learning processes. In this study, the StandardScaler method was employed to normalize the input features. StandardScaler transforms the data by subtracting the mean and scaling to unit variance, ensuring that each feature has a mean of zero and a standard deviation of one. This transformation was applied exclusively to the independent variables, preserving the original range of the target variable (WQI Value). Scaling was performed because the chosen ML models required it, as their feature distance-based operations and gradient optimization methods became significantly more effective. Properly scaling the data was necessary because it facilitated faster model convergence and enhanced learning accuracy.

Furthermore, principal component analysis (PCA) was performed to reduce the number of features and preserve most of the original data variance. Fig. 3 indicates that the top four principal components accounted for around 95% of all the variance, indicating that they can be used for the training phase. The change occurs within the fourth component that can be seen as a clear point of change in the cumulative explained variance plot and justifies the decision.

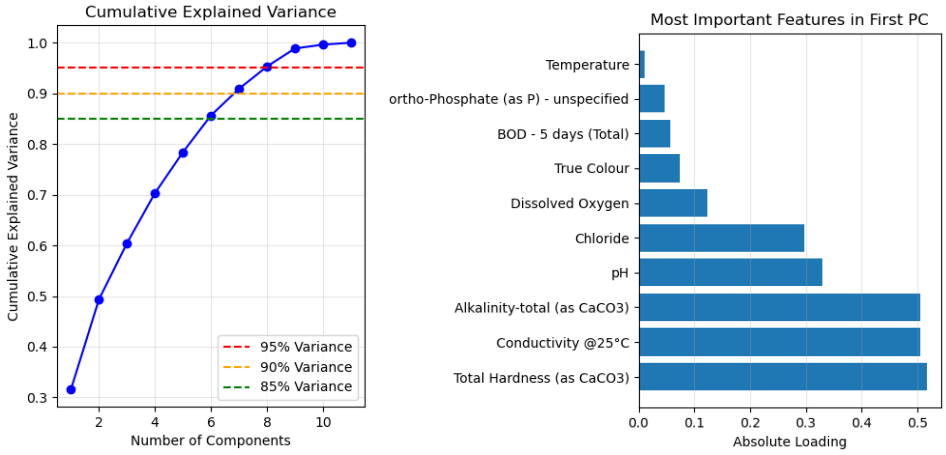

Fig. 3. PCA Visualization

The data in Fig. 3 clearly suggests that Total Hardness, Conductivity at 25°C, and Alkalinity are the dominant contributors to the first principal component. Additionally, PC2 was the most closely related to the target variable, WQI Value, so it may reveal important information for prediction. In short, using PCA reduced the number of dimensions in the data without losing key data details. The pipeline model was built using the top four components identified from PCA.

Lastly, the feature-selection process was used to identify correlations and reduce the number of features. Feature selection reduces the dimensionality of the training dataset by identifying and choosing only the features relevant to the target outcome (Pudjihartono et al., 2022). This process is crucial for creating a model that can generalize well when applied to new data. The correlation between each feature and the target, WQI, was studied by computing a Pearson correlation matrix. Any features found to have an absolute correlation coefficient over 0.5 were chosen as major predictors for WQI. Based on Fig. 4, Dissolved Oxygen turned out to be an important predictor due to its strong and negative correlation with the WQI ($r = -0.59$). This suggests that the model depends on dissolved oxygen because it substantially affects the WQI. A moderate negative correlation was found between True Colour and the WQI ($r = -0.52$), suggesting that changes in water colour also change how we predict water quality. The model excluded the variables Alkalinity, Total Hardness and Conductivity because they are somewhat related to the WQI,

https://doi.org/10.24191/mij.v7i1.11950

but both they and Total Solids are possible sources of multicollinearity. The model tries to improve its accuracy by emphasizing strong correlation between inputs and output.

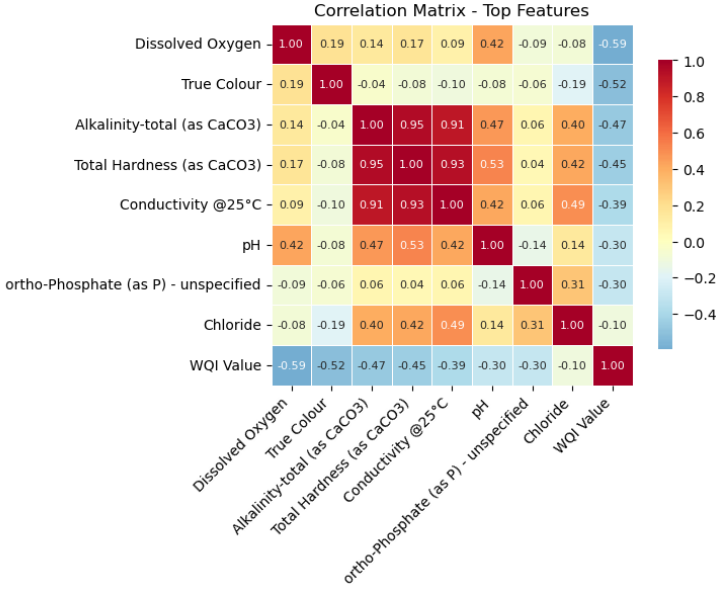

Fig. 4. Feature Correlation

2.3 Model Selection

The quality of predictions is heavily dependent on the selection of the algorithm. Therefore, six algorithms will be used in the analysis which are Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multi-Layer perceptron (MLP), Gradient Boosting (GB) and XGBoost. These algorithms have very specific strengths which depend on the data characteristics and prediction requirements. SVM draws a line through the data that gives the data some wiggle room within the epsilon margin. The kernel trick is applied in SVM to change ordinary inputs into a space with more dimensions. Commonly used kernels are linear, polynomial, RBF, and sigmoid (Schonlau, 2023). As a result, SVM can be reliably used to address difficult regression tasks. Meanwhile, DT model is a very versatile, nonparametric supervised learning technique that can be used for regression and classification problems. The process of generating a DT starts with the decision of which attribute should split the data, using such measures as Gini index of inequality, or entropy, or information gain. The dataset is then split into subgroups relative to the chosen attribute, then this process is done iteratively on each subgroup and generates new internal nodes or leaf. Splitting continues until one of the stopping conditions is reached, which is the depth limit or node homogeneity is achieved (Jain, 2016).

RF is a method based on DT, which constructs them on different bootstrap samples of the learning set, where instances are taken with replacement. At the time of training each tree, during the branching of tree, a few random features are chosen so that every tree in the ensemble learns different characteristics of data. The basic idea of after training is that the output of the algorithm involved is the total up of the outcomes of all the trees. In classification, the final prediction is made by the majority vote among the trees. The feature randomness together with the means employed in bagging make RF a highly accurate and relatively immune to overfitting (Salman et al., 2024). MLP is a three-layer type of artificial neural network. It is efficient in capturing complex patterns and can be scaled up and can handle the tasks of different

https://doi.org/10.24191/mij.v7i1.11950

complexity levels. But it takes more time to train or predict large-scale or high-dimensional data, and it depends heavily on activation function, architecture and hyperparameters (Wang et al., 2023). In addition, GB is also considered as one of the powerful ensemble methods for regression that builds a strong predictor by combining multiple weak learners, typically decision trees. It operates through an additive process, where each new tree is trained to minimize the residual errors of the previous model. GB combines with weak classification methods, usually DT, and consequently brings together vast numbers of them to make a strong classifier. The algorithm develops the model step by step as it does other boosting strategies, in that, in one step it aims at the instances that are mislabeled and in the next step assigns them higher weights (Shams et al., 2024).

Extreme Gradient Boosting (XGBoost) can be defined as an advanced ML algorithm that predicted to improve model performance based on GB and DT techniques. It comes with high performance and flexibility in its operation because it trains DT in sequentially to enhance the accuracy of its prediction following the errors made by the preceding tree. The guidance is formed to follow the negative of the gradient of the objective function (Wang et al., 2023). XGBoost is highly regarded for its speed and scalability, achieved through data structures that allow for parallel processing, efficient handling of missing data, and optimization of memory and speed. The algorithm performs well with complex data patterns due to its built-in regularization. Model training is conducted by default and fine-tuned using GridSearchCV to adjust parameters such as learning_rate, n_estimators, and max_depth to ensure better results and prevent overfitting.

2.4 Experimental Setup

The experimental phase of this study was conducted using the Python programming language within the VS Code integrated development environment. The development included the use of Scikit-learn and XGBoost, each being applied for ML as well as data preprocessing tasks, including running XGBoost. All computations and model training tasks were executed locally on a laptop using a Central Processing Unit (CPU) environment. The available hardware resources were sufficient to efficiently complete the training, evaluation, and visualization processes without the need for external cloud computing or GPU acceleration. The six ML models constructed in this study were the RF, MLP, DT, SVR, GB, and XGBoost. The selection of multiple algorithms allowed this research to evaluate distinct prediction methods for WQI. All models were evaluated using the same GridSearchCV hyperparameter search space, regardless of whether the data was normalized, feature-selected or reduced using PCA in the three training situations. In Table 2 shows each model's hyperparameters list. As a result of this method, performance is compared fairly, considering only data preprocessing steps.

Table 2. GridSearchCV Hyperparameter Settings

Model	Hyperparameters
XGBoost Regressor	learning_rate: [0.01, 0.05, 0.1, 0.2] n_estimators: [100, 300, 500, 700, 1000] gamma: [0.001]
Random Forest Regressor	n_estimators: [100, 300, 500, 700, 1000] max_depth: [None, 10, 20, 30]
Gradient Boosting Regressor	n_estimators: [100, 300, 500, 700, 1000] learning_rate: [0.01, 0.05, 0.1] max_depth: [3, 5, 7] min_samples_split: [2, 5] subsample: [0.8, 0.9, 1.0]
Decision Tree Regressor	max_depth: [None, 5, 10] min_samples_split: [2, 5, 10] min_samples_leaf: [1, 2, 4]

https://doi.org/10.24191/mij.v7i1.11950

Support Vector Regressor	kernel: ['linear', 'rbf'] C: [0.1, 1, 10, 100] epsilon: [0.01, 0.1, 0.5, 1.0] gamma: ['scale', 'auto', 0.01, 0.1, 1]
Multi-Layer Perceptron Regressor	activation: ['relu', 'tanh'] solver: ['adam', 'sgd'] alpha: [0.0001, 0.001, 0.01] learning_rate: ['constant', 'adaptive']

These hyperparameters were chosen according to the best practices established in previous studies and public official documentation in Scikit-learn and XGBoost. GridSearchCV was used to check how to tune each and every algorithm to attain the best configuration. These hyperparameters were mostly derived based on the official reference of Scikit-learn and XGBoost, and some of the previous research, like Shams et al. (2024) and Khaskheli et al. (2024), where hyperparameters settings were similar to those used to optimize the performance of the models. The description of each of the hyperparameters is given below:

(i) learning_rate (XGBoost, GB): The rate at which the model can learn each increment in the boosting process. The lower the values, the slower one learns, but it may lead to higher accuracy.

(ii) n_estimators (XGBoost, RF, GB): It is the number of trees to be constructed in the model. The performance can be enhanced using more trees but also consume more computational time.

(iii) gamma (XGBoost): Minimum loss decrease need to create a split. There are increased values which lead to a more sensitive algorithm.

(iv) max_depth (RF, GB, DT) The upper bound of the depth of each decision tree. Size of tree is also limited in order to avoid overfitting.

(v) min_samples_split (GB, DT): The minimum required number of samples that is sufficient in splitting an internal node. Regulates the extent of specific tree can be.

(vi) min_samples_leaf (DT): The number of samples needed to reach a leaf node. Assists in smoothing of the model.

(vii) subsample (GB): The proportion of samples put in the training of every tree. Footer on overfitting by bringing in randomness.

(viii) kernel (SVR): The kind of kernel function e.g. 'linear', 'rbf') that they agree to map the data to higher dimensionality.

(ix) C (SVR): The parameter which settles the conflict between smooth decision boundary and accurate classification of the training points.

(x) epsilon (SVR): regulates a range of tolerance, without applying penalties in case the errors occur.

2.5 Model Evaluation

Instead of depending on accuracy or precision, regression models are scored by examining their error and the variance in outcomes. The main project measurement ways are Root Mean Squared Error (RMSE), R-squared (R^2) and training time. The RMSE shows how big the average errors are, placing greater emphasis on the big ones and explains how accurate the predictions are compared to the actual results. R^2 should be near to 1, meaning the model fits the data of the target variable effectively. Finally training time, to check whether the model takes time to get the prediction value. The metrics are required to understand the accuracy of predictions for continuous results and compare the results of different regression algorithms. Furthermore, training time is required to evaluate the computational efficiency of each of the models. This

https://doi.org/10.24191/mij.v7i1.11950

allows comparisons to be made not only based on the resulting predictive accuracy, but also with respect to the time it would take to train the model which would be of great concern when dealing with large amounts of data, or when the model needs to be retrained frequently. RMSE describes the average difference between the real values and values that have been predicted. This is done by taking square root of the mean of the squared differences between squared values of the predicted and the true values. RMSE can easily show how wrong the predictions are since they are expressed in the same units as the target variable. The lower the value of RMSE, the better the model fits.

$$\text{RSME} = sqrt\ (MSE) \tag{3}$$

The R^2 (Coefficient of Determination) reveals how good the predictions of the model compare with the real ones. It provides us with the value of the extent to which the model can explain the difference between the actual WQI values. R^2 value approximating 1 indicates that the model is optimal, and an R^2value such that is near 0 implies that the model poorly fits the data.

$$\text{R}^2\ \text{Score} = \frac{\Sigma(\text{ypred} - \text{ytrue})^2}{\Sigma(\text{ytrue} - \text{ymean})^2} \tag{4}$$

Training time is the time used to learn by the training data on the model. It can be used to assess the efficiency of the model in case very large amounts of data are processed, or when one is required to achieve results in a short time.

3. RESULTS AND DISCUSSION

The training procedure followed an initial default parameter model development stage followed by GridSearchCV-based performance enhancement stage. Both steps formed an approach to create an initial performance benchmark followed by optimized results through controlled hyperparameter manipulation. Prior to training, feature scaling was applied on the range of input features using StandardScaler. This was significant considering the models that responded to the size of the features like the SVR and the MLP. RF, DT, GB and XGBoost are all models which use the tree-based mode of logic and do not require the feature scaling since it did not impact the function of the model.

3.1 Model Training on PCA-Reduced Data

In this phase, PCA was applied to reduce the number of dimensions. Before turning the data into principal components, StandardScaler was first used to standardize everything. With PCA, the top four components were selected, as they together explained about 85% of what the dataset contained. In case PCA is applied to the original input features, then SVR, GB and MLP can be used, as the resulting features are fewer and not correlated. Tree-based models such as DT and RF do not work well with PCA because tree-based models require the original feature ordering to understand rules.

3.2 Model Training using Normalized Data

In this phase, all regression models were trained using the dataset after applying normalization via the Standard Scaler method. As a result, all the features in the dataset each had a mean of zero and a standard deviation of one. Both SVR and MLP depend on distance and gradients, so applying normalization is necessary to avoid their sensitivity to feature magnitude. All models were trained on the normalized dataset using an 80:20 train-test split. Hyperparameter tuning was performed using GridSearchCV with the consistent parameter ranges shown earlier in Table 2. The model was tuned with 5-fold cross-validation to

https://doi.org/10.24191/mij.v7i1.11950

guarantee it would not overlearn and give better reliability. While normalization made little difference to DT, RF, GB and XGBoost, it highly improved results for SVR and MLP. The results were evaluated using RMSE and R^2 score. This section presents the performance outcomes of ML regression models trained on the normalized dataset only, following the removal of outliers and application of Standard Scaler normalization. This preprocessing step was critical for models that rely on feature scale, particularly SVR and MLP, which are sensitive to input magnitudes. As shown in Table 3, XGBoost delivered the best baseline accuracy, with an R^2 of 0.9901 and an RMSE of 0.7200. This was followed closely by DT and RF, both achieving strong R^2 values of 0.9867 and 0.9862, with RMSEs of 0.8354 and 0.8491, respectively.

Table 3. Model Performance on Normalized Data Only (Baseline)

Model	R^2	RMSE	Time (s)
XGBoost	0.9901	0.7200	0.22
RF	0.9862	0.8491	2.76
GB	0.9606	1.4358	0.82
DT	0.9867	0.8354	0.04
SVR	0.9050	2.2308	0.99
MLP	0.9320	1.8874	5.56

Table 4 highlights the optimized performance. GridSearchCV showed that GB performed best, featuring a maximum R^2 of 0.9943 and the lowest RMSE at 0.5457. These results demonstrate the value of adjusting multiple models with data after normalization. Even with optimization, XGBoost achieved very similar results, with R^2 of 0.9940 and RMSE of 0.5618. Before optimization, SVR managed an R^2 of 0.9050 and an RMSE of 2.2308, but improved after being optimized, reaching an R^2 of 0.9516 and RMSE of 1.5923. Normalization and hyperparameter optimization had a positive effect, as the MLP model results improved from R^2 of 0.9320 to 0.9508 and RMSE from 1.8874 to 1.6056. In terms of training time, the Gradient Boosting and MLP models showed the highest computational cost after optimization, with training times of approximately 10.98 seconds and 10.32 seconds, respectively. In conclusion, normalization helped scale-sensitive models learn faster and prevented training instability. The test revealed that GB, when trained on normalized data and correctly tuned, consistently delivered the best and most accurate WQI estimations in the configuration tested.

Table 4. Model Performance on Normalized Data Only (Optimized)

Model	R^2	RMSE	Time (s)
XGBoost	0.9940	0.5618	0.67
RF	0.9860	0.8549	13.23
GB	0.9943	0.5457	10.98
DT	0.9867	0.8343	0.04
SVR	0.9516	1.5923	7.69
MLP	0.9508	1.6056	10.32

3.3 Model Training using Feature Selected Data

This experiment used a reduced dataset by selecting the most relevant features based on correlation analysis. According to Figure 3, features with a strong linear relationship with the WQI specifically those with an absolute Pearson correlation coefficient above 0.5 were retained. These included Dissolved Oxygen and True Colour. Features with weaker correlations, such as Alkalinity, Total Hardness, Conductivity, pH, Orthophosphate, Chloride, Ammonia, BOD, and Temperature, were excluded. Overall, the results showed that feature selection led to improved efficiency and slightly faster training convergence. Both SVR and DT worked much better when using simplified data. Performance remained good for both RF and XGBoost,

https://doi.org/10.24191/mij.v7i1.11950

demonstrating they are resistant to using fewer features. This section presents the performance outcomes of ML regression models trained in a reduced set of features, selected based on their correlation with the WQI target variable. The selection process was performed to retain only the most relevant inputs and improve model efficiency.

As shown in Table 5, the RF recorded the best baseline performance, achieving an R^2 of 0.8394 and RMSE of 2.9002, followed closely by the DT, with an R^2 of 0.8322 and RMSE of 2.9646. Despite using fewer features, these tree-based methods continued to perform well for prediction. GB and XGBoost showed relatively lower baseline performance with R^2 values of 0.6783 and 0.7882, indicating reduced effectiveness with limited feature sets.

Table 5. Model Performance on Feature Selected (Baseline)

Model	R^2	RMSE	Time (s)
XGBoost	0.7882	3.3309	0.07
RF	0.8394	2.9002	0.81
GB	0.6783	4.1052	0.24
DT	0.8322	2.9646	0.01
SVR	0.6318	4.3914	0.92
MLP	0.6288	4.4094	2.18

Table 6 displays the optimized results. After applying GridSearchCV, GB emerged as the best-performing model, reaching a maximum R^2 of 0.8621 and the lowest RMSE of 2.6879. Both XGBoost and RF also showed strong improvements, with R^2 values of 0.8449 and 0.8435, and RMSEs of 2.8502 and 2.8631, respectively. The results suggest that hyperparameter tuning plays a big part in boosting performance, regardless of how simple the input data is. Before optimization, SVR and MLP models performed poorly, with R^2 values of 0.6318 and 0.6288. After tuning, minor improvements were observed: SVR reached an R^2 of 0.6378, and MLP reached 0.6439, with RMSEs of 4.3555 and 4.3191, respectively. This indicates that these models struggle to generalize well when trained on fewer features. After optimization, some of the models spent more time learning. RF required approximately 7.27 seconds, while GB took 2.30 seconds. In contrast, XGBoost remained highly efficient, completing training in just 0.42 seconds. MLP again showed a high computational cost with an optimized training time of 5.45 seconds. As a result, utilizing only the most important features discovered that GB models remain both accurate and reliable compared to others. Although models with a few features might perform poorly, collecting datasets using multiple techniques still helps with WQI estimates in areas with few features.

Table 6**.** Model Performance on Feature Selected (Optimized)

Model	R^2	RMSE	Time (s)
XGBoost	0.8449	2.8502	0.42
RF	0.8435	2.8631	7.27
GB	0.8621	2.6879	2.30
DT	0.8322	2.9646	0.01
SVR	0.6378	4.3555	0.73
MLP	0.6439	4.3191	5.45

3.4 Model Training on PCA-Reduced Data

This section looks at the performance of ML regression models built using features reduced with PCA. The data was reduced in dimension through PCA, while about 95% of its total variance stayed the same. The purpose was to select just the important input variables while making sure the majority of the important

https://doi.org/10.24191/mij.v7i1.11950

WQI details were still included. Table 7 displays baseline model performance using PCA-transformed features. Among all models, XGBoost delivered the best baseline results, achieving an R^2 of 0.9700 and an RMSE of 1.2527. This was followed closely by RF R^2 of 0.9656 and RMSE of 1.3424 and Decision Tree R^2 of 0.9612 and RMSE of 1.4256. GB and SVR both produced good outcomes, with R^2 scores of 0.8988 and 0.9003 and MLP achieved the best results with an R^2 of 0.9233 and an RMSE of 2.0049.

Table 7. Model Performance on PCA (Baseline)

Model	R^2	RMSE	Time (s)
XGBoost	0.9700	1.2527	0.10
RF	0.9656	1.3424	3.04
GB	0.8988	2.3021	1.20
DT	0.9612	1.4256	0.04
SVR	0.9003	2.2852	1.08
MLP	0.9233	2.0049	5.54

The above Table 8 displays the hyperparameter tuning using GridSearchCV and the models generally showed improved performance. The R^2 of 0.9795 for GB made it the best in the group, along with the lowest RMSE score of 1.0366. Even after reduced the data's dimensions, XGBoost was able to perform well, showing an R^2 of 0.9793 and RMSE of 1.0407. RF achieved a slight improvement with R^2 of 0.9668 and RMSE of 1.3195. SVR and MLP also saw modest gains. SVR improved to an R^2 of 0.9452 with an RMSE of 1.6948, and MLP reached an R^2 of 0.9320 and RMSE of 1.8869. Meanwhile, the Decision Tree model did not benefit from optimization, maintaining its R^2 of 0.9612, which suggests a performance plateau under the PCA configuration. Some models had higher computational expenses due to optimization changes in their training time. RF needed 20.74 seconds to finish, while GB took just 18.24 seconds. In contrast, XGBoost finished in only 0.69 seconds and showed that it is scalable. SVR took 8.47 seconds to run and MLP used 7.07 seconds. In conclusion, running PCA with 95% retained variance made it possible for most models to keep an accurate prediction with less complicated data. GB was found to be the most accurate, with XGBoost very close behind, so both could be applied to WQI prediction with less input data.

Table 8. Model Performance on PCA (Optimized)

Model	R^2	RMSE	Time (s)
XGBoost	0.9793	1.0407	0.69
RF	0.9668	1.3195	20.74
GB	0.9795	1.0366	18.24
DT	0.9612	1.4256	0.04
SVR	0.9452	1.6948	8.47
MLP	0.9320	1.8869	7.07

3.5 Discussion

This section explores a comparison among optimized ML models implemented using three data preprocessing methods: normalization, feature selection and PCA. Evaluation is done using three factors, which is R^2 for how well predicts, RMSE for how far the predictions vary from actual values and the number of seconds it takes to learn. The performance trends of the top three models are GB, XGBoost, and RF are shown in Fig. 5, which highlights their behavior across the tested preprocessing methods. Among all configurations, the model trained on normalized data produced the most accurate results. GB outperformed the other algorithms, having an R^2 value of 0.9943 and the RMSE of just 0.5457. XGBoost performed very closely, reporting R^2 of 0.9940 and RMSE of 0.5618, joined by RF at an R^2 of 0.9860 and RSME of 0.8549. While the GB model was less precise, XGBoost trained much more quickly, finishing in just 0.67 seconds,

https://doi.org/10.24191/mij.v7i1.11950

against the GB's 10.98 seconds. These findings indicate that GB outperforms others for maximum accuracy, but if time is important, XGBoost is the best.

After using feature-selected data in training, the tree-based models lost some of their accuracy, though it still worked well. GB again outperformed the others with an R^2 of 0.8621 and RMSE of 2.6879. Both XGBoost and RF recorded similar R^2 values around 0.844, with RMSEs just above 2.85. These findings indicate that tree-based ensemble methods can generalize reasonably well even with a reduced set of input features. Notably, XGBoost remained the most efficient, completing training in only 0.42 seconds, reinforcing its suitability for time-sensitive applications. In the PCA-reduced dataset configuration, GB continued to lead, achieving an R^2 of 0.9795 and RMSE of 1.0366. XGBoost closely followed with an R^2 of 0.9793 and RMSE of 1.0407, demonstrating that both models retained strong predictive power even after dimensionality reduction. RF achieved an R^2 of 0.9668 but exhibited a much longer training time of 20.74 seconds. These results confirm that PCA effectively preserves essential data variance while simplifying the model's input structure, allowing robust performance from complex models. It is important to note that tree-based models like GB, XGBoost, and RF are naturally resistant to feature scaling due to their reliance on threshold-based data splitting rather than absolute feature magnitudes. As a result, these models consistently performed well across all preprocessing types.

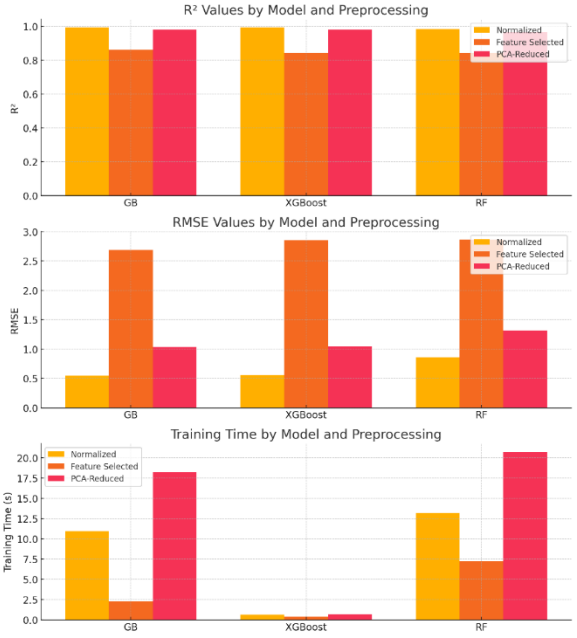

Fig. 5. Comparison Between Three Method

https://doi.org/10.24191/mij.v7i1.11950

4. CONCLUSION

After applying all the techniques to the testing and training, only feature normalization managed to result in the highest accuracy of all evaluated models consistently. Both techniques for dimensionality reduction, which are feature selection and PCA, did not outperform the normalized dataset in terms of accuracy or how close the predictions were to the actual values. This points out that scaling is important, especially in models that use neural networks such as SVM and MLP. GB, RF and XGBoost were identified as the top-performing models, because it had the highest R^2 values and the smallest RMSE on the normalized dataset. It was shown that GB algorithm is accurate and can be deployed to check water quality real time. In the end, the testing proved that, when preprocessing and adjusting parameters are done well, ML models deliver accurate and prompt predictions for making timely decisions. However, this result is only applicable to the specific data that has been used in this study. Future need should be to continue to collect water quality data from around a variety of different geographic regions to build models that are more adaptable and reliable in different environments. Due to local pollution sources, climate condition and human activity, water sources have different characteristics at different sites.

5. ACKNOWLEDGEMENT/ FUNDING

The authors would like to acknowledge the support given by the Faculty of Data Science and Computing, Universiti Malaysia Kelantan on the conduct of this research.

6. CONFLICT OF INTEREST STATEMENT

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare the absence of conflicting interests with the funders.

7. AUTHORS' CONTRIBUTIONS

Muhamad Danish Saiful Rizal led the research execution, data analysis, and manuscript drafting. Nurzulaikha Abdullah contributed to the development of the theoretical framework and assisted in refining the analysis. Mohd Hakimi Aiman Ibrahim was responsible for literature review and manuscript review. Fakhitah Ridzuan supervised the research, provided overall guidance throughout the project, and conducted the final review and approval of the manuscript for submission.

8. DECLARATION OF GENERATIVE AI IN THE WRITING PROCESS

The authors declare that generative artificial intelligence (AI) tools may have been used solely to support language refinement during the preparation of this manuscript. Such tools were used only to improve grammar, sentence structure, clarity, and overall readability. All aspects of the research, including the study design, data collection, data extraction, analysis, interpretation of findings, and final academic decisions, were conducted, reviewed, and approved by the authors. The authors take full responsibility for the accuracy, integrity, originality, and scholarly content of this manuscript.

9. DATA AVAILABILITY/SUPPLEMENTARY MATERIALS

The data and/or supplementary materials supporting the findings of this study are available from the corresponding author upon reasonable request, where applicable.

https://doi.org/10.24191/mij.v7i1.11950

10. ETHICS STATEMENT

The authors declare that this study did not involve human participants or animal subjects. Where applicable, all procedures were conducted in accordance with relevant institutional guidelines, safety requirements, and ethical standards.

REFERENCES

Bala, B., & Behal, S. (2024). A Brief Survey of Data Preprocessing in Machine Learning and Deep Learning Techniques. 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 1755–1762. https://doi.org/10.1109/I-SMAC61858.2024.10714767

Chidiac, S., El Najjar, P., Ouaini, N., El Rayess, Y., & El Azzi, D. (2023). A comprehensive review of water quality indices (WQIs): history, models, attempts and perspectives. Reviews in Environmental Science and Biotechnology, 22(2), 349–395. https://doi.org/10.1007/S11157-023-09650-7/TABLES/7

Dhawas, P., Dhore, A., Bhagat, D., Pawar, R. D., Kukade, A., & Kalbande, K. (2024). Big data preprocessing, techniques, integration, transformation, normalisation, cleaning, discretization, and binning. In D. Darwish (Ed.), *Big Data Analytics Techniques for Market Intelligence* (pp. 159–182). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-0413-6.ch006

European Commission, Directorate-General for Environment. (2025, January 16). *Global urbanisation identified as the landscape change most responsible for water-quality deterioration over last 20 years*. European Commission.

Jain, R. K., Natarajan, R., & Ghosh, A. (2016). Decision Tree Analysis for Selection of Factors in DEA: An Application to Banks in India. *Global Business Review*, *17*(5), 1162-1178. https://doi.org/10.1177/0972150916656682

Jansen, B. J., Aldous, K. K., Salminen, J., Almerekhi, H., & Jung, S. (2024). Data preprocessing. In *Understanding Audiences, Customers, and Users via Analytics* (pp. 65–75). Springer. https://doi.org/10.1007/978-3-031-41933-1_6

Kang, M., & Tian, J. (2018). Machine Learning: Data Pre-processing. Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things, 111–130. https://doi.org/10.1002/9781119515326.CH5

Khaskheli, A. S., Ahmed Rahu, M., Siraj, S., Jamshed, H., Iqbal, S., & Shah, N. (2024). Optimized water quality forecasting using machine learning. *International Journal of Information Systems and Computer Technologies, 3*(2), 46–60. https://doi.org/10.58325/IJISCT.003.02.0094

Kim, H. Il, Kim, D., Mahdian, M., Salamattalab, M. M., Bateni, S. M., & Noori, R. (2024). Incorporation of water quality index models with machine learning-based techniques for real-time assessment of aquatic ecosystems. Environmental Pollution, 355, 124242. https://doi.org/10.1016/J.ENVPOL.2024.124242

Li, G., Li, J., & Mitrouchev, P. (2023). A Data Preprocessing Method for Strip Steel. In Y. Wang, T. Yu, & K. Wang (Eds.), Advanced Manufacturing and Automation XII (pp. 391–398). Springer Nature Singapore.

https://doi.org/10.24191/mij.v7i1.11950

Montgomery, D. C., & Runger, G. C. (2010). *Applied statistics and probability for engineers* (5th ed.). John Wiley & Sons.

Ozsahin, D. U., Taiwo Mustapha, M., Mubarak, A. S., Said Ameen, Z., & Uzun, B. (2022). Impact of feature scaling on machine learning models for the diagnosis of diabetes. Proceedings - 2022 International Conference on Artificial Intelligence in Everything, AIE 2022, 87–94. https://doi.org/10.1109/AIE57029.2022.00024

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., & O'Sullivan, J. M. (2022). A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. Frontiers in Bioinformatics, 2, 927312. https://doi.org/10.3389/FBINF.2022.927312/FULL

Rahman, A., Syeed, M. M., Fatema, K., Karim, M. R., Khan, R. H., Hossain, M. S., & Uddin, M. F. (2025). *Surface Water Quality Index (WQI) Forecasting Dataset* (Version 1) [Data set]. Figshare. https://doi.org/10.6084/m9.figshare.28184252.v1

Salman, H., Kalakech, A., & Steiti, A. (2024). Random Forest Algorithm Overview. Babylonian Journal of Machine Learning, 2024, 69–79. https://doi.org/10.58496/BJML/2024/007

Schonlau, M. (2023). Support vector machines. In *Applied Statistical Learning: With Case Studies in Stata*. Springer. https://doi.org/10.1007/978-3-031-33390-3

Shams, M. Y., Elshewey, A. M., El-kenawy, E. S. M., Ibrahim, A., Talaat, F. M., & Tarek, Z. (2024). Water quality prediction using machine learning models based on grid search method. Multimedia Tools and Applications, 83(12), 35307–35334. https://doi.org/10.1007/S11042-023-16737-4/FIGURES/15

Sierra-Porta, D. (2024). Assessing the impact of missing data on water quality index estimation: a machine learning approach. Discover Water 2024 4:1, 4(1), 1–20. https://doi.org/10.1007/S43832-024-00068-Y

Wang, X., Li, Y., Qiao, Q., Tavares, A., & Liang, Y. (2023). Water Quality Prediction Based on Machine Learning and Comprehensive Weighting Methods. Entropy 2023, Vol. 25, Page 1186, 25(8), 1186. https://doi.org/10.3390/E25081186

Yasnitsky, L., & Plotnikova, E. (2024). A neural network algorithm for identifying and removing outliers in noisy data sets. *Journal of Applied Informatics, 19*(5), 88–100. https://doi.org/10.37791/2687-0649-2024-19-5-88-100

Zhu, M., Wang, J., Yang, X., Zhang, Y., Zhang, L., Ren, H., Wu, B., & Ye, L. (2022). A review of the application of machine learning in water quality evaluation. Eco-Environment and Health, 1(2), 107–116. https://doi.org/10.1016/J.EEHL.2022.06.001

https://doi.org/10.24191/mij.v7i1.11950