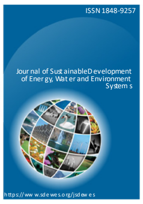
Journal of Sustainable Development of Energy, Water and Environment Systems

http://www.sdewes.org/jsdewes

Year 2026, Volume 14, Issue 1, 1130634



Original Research Article

Machine Learning-Based Water Quality Prediction

Ali Al-Ataby[*], *Beza Negash Getu*, *Hussain Attia*
Electrical and Electronics Engineering Department, American University of Ras Al Khaimah,
Ras Al Khaimah, United Arab Emirates
e-mail: ali.ataby@aurak.ac.ae, bgetu@aurak.ac.ae, hattia@aurak.ac.ae

Cite as: Al-Ataby A., Negash Getu B., Attia H., Machine Learning-Based Water Quality Prediction, J.sustain. dev. energy water environ. syst., 14(1), 1130634, 2026, DOI: https://doi.org/10.13044/j.sdewes.d13.0634

ABSTRACT

Water is an indispensable resource for all forms of life, with a particularly critical role in supporting human health, agriculture, and industrial development. With the predicted water scarcity worldwide, it is crucial to have a tool to analyse and predict water potability accurately and in real-time. This study used machine learning models to predict water potability based on quality features such as potential of hydrogen (pH) value, hardness, solids content, chloramines, sulfate, and conductivity. Potability is determined based on the concentration of these features in the water. Four Machine Learning algorithms, namely, Random Forest (RF), Logistic Regression, Extreme Gradient Boosting (XGBoost), and Deep Learning Neural Networks, are used to analyse water potability after training using a water quality dataset. Initial experiments showed moderate performance, with RF (F1-score = 0.47 and area under the receiver operating characteristic curve of 0.68) and XGBoost (F1-score = 0.49 and area under the receiver operating characteristic curve of 0.66), outperforming the other two models. After addressing class imbalance and introducing more features using feature engineering, the performance of the four models was significantly improved, with RF achieving an F1-score of 0.85 and an area under the curve of 0.90 and XGBoost achieving an F1-score of 0.86 and an area under the curve of 0.91. The results clearly indicate that Random Forest and XGBoost consistently outperformed the Linear Regression model and the Deep Learning model in terms of predictive accuracy and robustness. These results demonstrate the critical importance of feature engineering and hyperparameter optimisation in enhancing model effectiveness. A real-time application for water potability prediction was developed to classify water as either “safe to drink” or “unsafe to drink”. Its functionality was successfully validated, and its output was displayed on a user-friendly graphical user interface.

KEYWORDS

Water, Potability, Machine Learning, Random Forest, XGBoost, Deep Learning, Feature Engineering, AUC.

INTRODUCTION

Water is one of the most critical natural resources, essential for life sustainability, supporting ecosystems, and enabling socio-economic development [1]. As the population grows worldwide, and urbanisation and industrialisation accelerate, the demand for clean and safe water has significantly increased [2]. However, the quality of water and its sources across the globe has been deteriorating due to pollution from domestic sewage, industrial discharge, agricultural activities and runoff, human-induced factors, and rapid urbanisation [3]. As a result of water pollution, human beings have begun to suffer from a variety of health problems, including skin disease, diarrhoea, dysentery, respiratory illnesses, anaemia, complications in

[*] Corresponding author

childbirth, and other health issues [4]. As water pollution increases, real-time monitoring and accurate prediction of water quality have become essential to ensure public health, environmental protection, and regulatory compliance [5]. Traditional water quality monitoring methods often rely on manual sampling and laboratory analyses, and are time-consuming, labour-intensive, and may not provide real-time results about water quality [6]. The traditional approaches involve laboratory testing using chemical or biological methods to measure parameters such as pH, hardness, sulfates, chloramines, solids turbidity, and others. Sensor technologies and the Internet of Things (IoT) have significantly improved data collection in water monitoring systems [7].

Nevertheless, as the need for precision and accuracy increases, most data generated from many sensors require robust computational models for real-time interpretation and actionable decision-making [8], [9]. Moreover, the increased complexity and volume of water quality data often collected through real-time sensors may necessitate advanced data-driven approaches to ensure timely and accurate assessment. In response to these challenges, the integration of Machine Learning (ML) techniques into water quality analysis and prediction has gained significant attention, offering the potential for efficient, accurate, and real-time monitoring solutions [10], [11].

This work employs a number of ML models to assess the quality of water and predict its potability. A dataset available online is used to train the developed models. Four ML algorithms, namely, Random Forest (RF), XGBoost, Logistic Regression (LR), and Deep Learning (Multilayer Perceptron or MLP), will be employed to assess and predict water potability for the given dataset. Performance was compared based on metrics such as precision, accuracy, recall, F1-score, and AUC (area under the receiver operating characteristic curve).

The contributions of this study are:

- Feature selection via importance analysis: A feature importance analysis was used to select a smaller number of impactful features to build the ML models. This approach makes the models lightweight with reduced computational overhead, and hence, more suitable for real-time deployment.
- Performance enhancement through preprocessing: The performance of the models was enhanced using class balancing, hyperparameter tuning, and feature engineering.
- Real-Time application deployment and graphical user interface (GUI) integration for practical use: The best performing model was integrated into a user-friendly water potability prediction application to classify water as either “safe to drink” or “unsafe to drink” in real-time.

The rest of this paper is structured as follows. Section II provides a literature review of the subject. The methods are discussed in Section III. Section IV provides simulation results and discussion. Conclusions are highlighted in Section V.

LITERATURE REVIEW

ML techniques are powerful tools for analysing complex, multivariate, multi-dimensional, and nonlinear datasets. ML algorithms, including supervised, unsupervised, and reinforcement learning paradigms, have proven to be powerful in modelling complex, nonlinear relationships, which is the case with environmental data [12], [13]. Supervised learning models such as Support Vector Machines (SVM), RF, and Artificial Neural Networks (ANN) have been employed to predict various water quality parameters, including pH, dissolved oxygen, turbidity, and biochemical oxygen demand [14], [15]. For instance, the work in [16] utilised supervised ML techniques to predict water quality parameters to achieve high accuracy levels. The study applied these models, revealing their superiority over conventional techniques in capturing nonlinear relationships between water parameters. The paper in [17] demonstrated the application of unsupervised ML for anomaly detection in water treatment systems to ensure water safety.

Many studies have also validated the capability of supervised learning algorithms such as ANNs and SVMs to predict water quality. The researcher in [18] performed a study to compare ML models and emphasised the role of big data in identifying water quality.

The utilisation of Deep Learning (DL) has further enhanced the capabilities of ML in water quality analysis. DL offers significant advantages in handling complex and high-dimensional water quality data. For example, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have been effective in modelling temporal dependencies in water quality time series data and generating accurate water quality prediction results, as shown in [19] and [20]. The study in [21] employed Convolutional Neural Networks (CNN) with LTSM (or CNN-LSTM) to simulate parameters such as pH and dissolved oxygen, showing improved accuracy in predicting water quality dynamics.

Hybrid approaches have gained attention for their enhanced predictive power. The researchers in [22] proposed a hybrid decision tree model that outperformed standalone algorithms in short-term water quality forecasting. The study in [23] demonstrated the effectiveness of integrating CNN with LSTM networks to predict short-term fluctuations in water quality. These models excel at modelling temporal and spatial dependencies in water quality datasets, especially under varying environmental conditions, and support the development of real-time water quality monitoring systems. The study in [24] used novel hybrid algorithms to improve water quality indices, while the analysis in [25] demonstrated the feasibility of real-time classification of water quality classes using supervised ML. Such systems are capable of making real-time decisions and provide an early-warning mechanism for water quality management.

Moreover, the integration of IoT devices with ML models has enabled the development of intelligent water quality monitoring systems. IoT sensors facilitate real-time data collection, which, when analysed with ML algorithms, can provide timely insights into water quality. The work in [26] proposed a probabilistic ML model integrated with IoT sensors for water quality level estimation, demonstrating its effectiveness in real-world scenarios. ML remains a powerful tool for generating predictions and trends and providing a comprehensive understanding and solution to complex problems and systems.

Identifying the most relevant features that are required by the ML models is essential for building efficient and interpretable models. The studies in [18] and [27] emphasised the use of data-driven techniques for selecting relevant features and pollution sources. Recent studies, e.g., [28], have explored interpretable ML models to quantify the effect of multiple pollutants on water quality prediction.

Despite these advancements, challenges exist in the application of ML to water quality analysis. High-quality and comprehensive datasets remain a prerequisite for practical model training. Issues such as data scarcity, sensor reliability, and model interpretability need to be addressed to fully utilise the power of ML in this domain [29]. Ongoing research focuses on developing robust, scalable, and interpretable ML models that can operate effectively under varying environmental conditions.

ML models have transformed the field of water quality analysis and prediction by offering scalable, fast, and adaptive solutions. From classical models such as SVMs to more advanced architectures such as CNN-LSTM hybrids, these methods have shown strong potential in prediction, classification, and real-time analysis. However, a number of challenges persist. First, data quality issues such as missing values and class imbalance are observed in environmental datasets. Although some studies acknowledge these problems, they are not usually treated systematically. Second, generalisability is limited in most of the studies that rely on a dataset from a single water source or region. Hence, it is challenging to use developed models with other sources or in different locations. Third, interpretability is important, particularly in environmental applications. Yet, DL models in the literature may not necessarily be transparent, with only a few studies utilising interpretable ML or feature importance analysis [28].

This study closes these gaps by handling missing values, applying class balancing, and using feature engineering and feature importance to improve the performance and interpretability of the models. Furthermore, comparing different models indicates the relative robustness of ensemble, linear, and neural network models [23], [24]. Incorporation of additional sources of information, such as IoT sensors and climate models, and building hybrid models for varying environmental conditions, are prospects to increase the accuracy of predictions and real-time monitoring [26].

METHODS

This section describes the experimental steps followed in this study, including dataset analysis and processing, performance metrics, model development, and deployment.

Dataset Analysis and Preprocessing

The water potability dataset was retrieved from an open-source repository [30]. It contains a total of 3,276 records, with 1,998 records labelled as non-potable or unsafe for drinking and 1,278 records labelled as potable or safe for drinking. The dataset includes nine features: pH, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity. Potability is represented by binary values, where 1 is for potable water and 0 is for non-potable water. These classifications are based on the concentration levels of the aforementioned substances and features.

The dataset was studied and investigated thoroughly to check its quality. It was found that there are 491 missing pH values, 781 missing sulfate values, and 162 missing trihalomethane values. The missing parameter values were replaced with the mean values, which is a common method, especially when the data are fairly symmetric.

The next step was to carry out exploratory data analysis (EDA). It includes examining histogram distribution for each feature, handling missing values (either by filling or removing), generating correlation analysis (heatmap), checking class balance for potability, and generating statistical summaries by calculating the mean, median, standard deviation, and minimum and maximum values. The results of this analysis were then presented visually. Figure 1 to Figure 4 show the distributions of pH, hardness, solids, and sulfate. From those figures, it can be seen that features are not perfectly normal, but many are skewed (e.g., solids, sulfate).

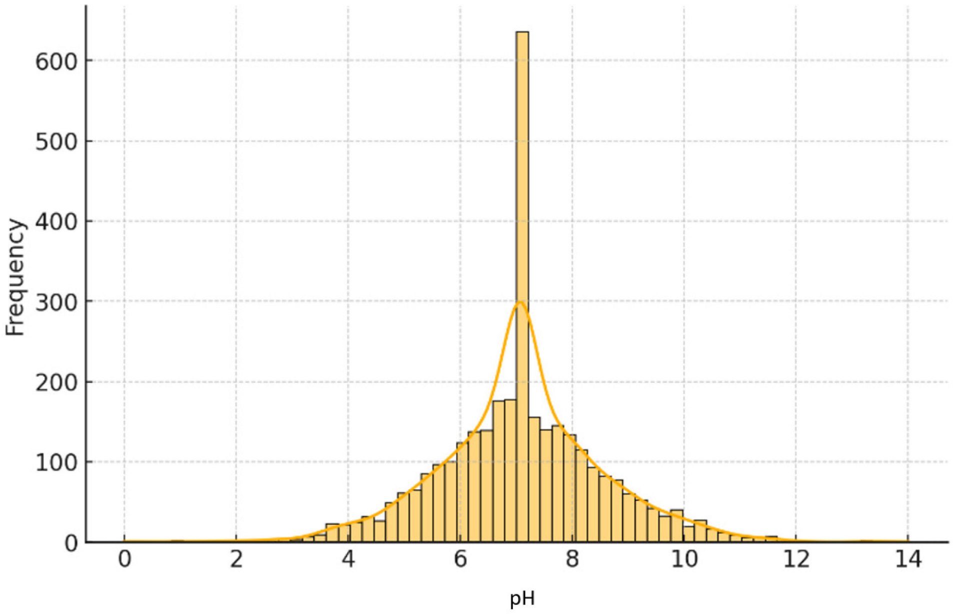


Figure 1. Distribution of pH

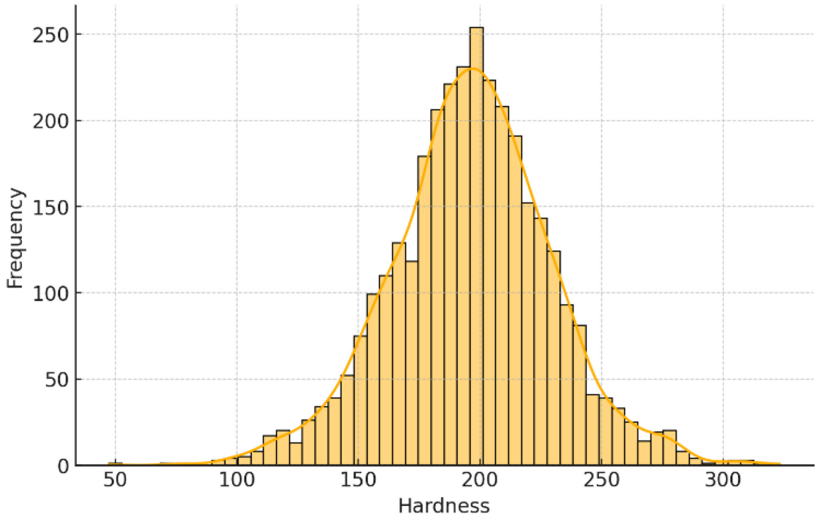


Figure 2. Distribution of hardness [mg/L]

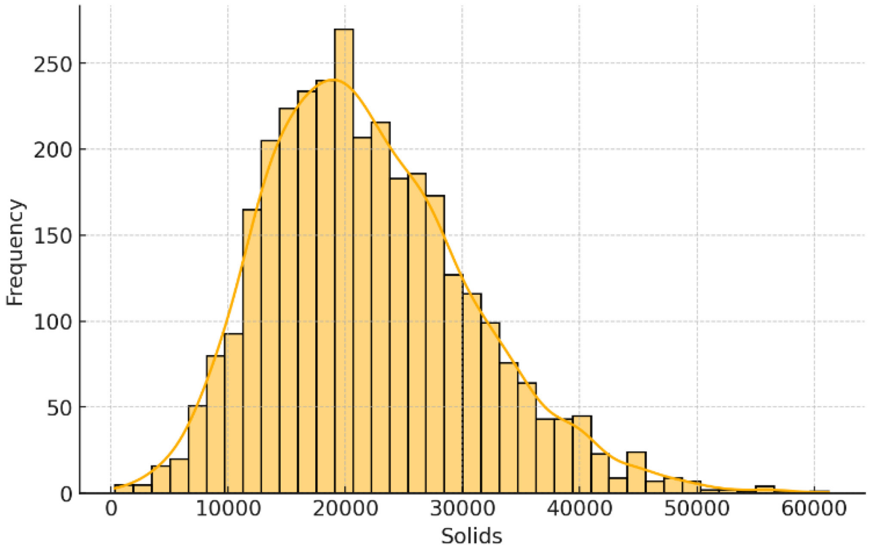


Figure 3. Distribution of solids [ppm]

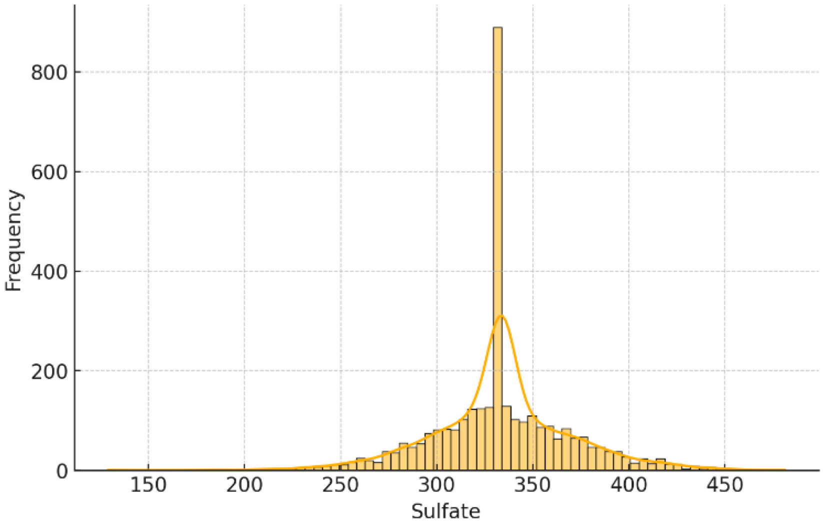
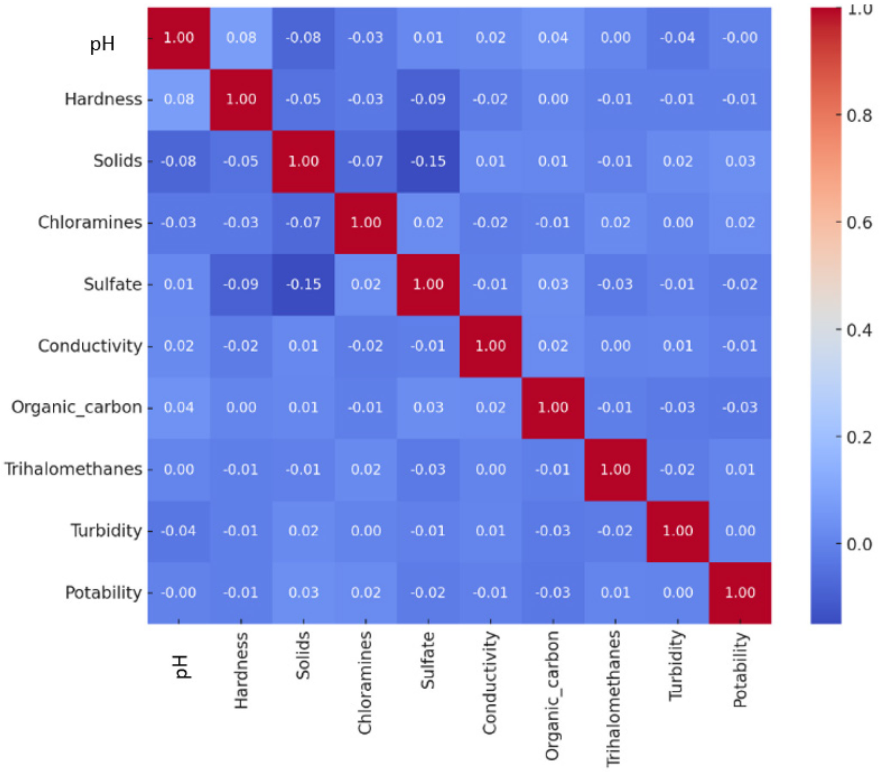


Figure 4. Distribution of sulfate [ppm]

Figure 5 shows the correlation heatmap of the nine features besides the potability. It can be seen that some positive correlations were observed (e.g., solids and conductivity). Most other features are weakly correlated with each other and with potability. In general, the weak correlation between features means the features are independent, and each feature has its own influence on the result of potability.



	pH	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
pH	1.00	0.08	-0.08	-0.03	0.01	0.02	0.04	0.00	-0.04	-0.00
Hardness	0.08	1.00	-0.05	-0.03	-0.09	-0.02	0.00	-0.01	-0.01	-0.01
Solids	-0.08	-0.05	1.00	-0.07	-0.15	0.01	0.01	-0.01	0.02	0.03
Chloramines	-0.03	-0.03	-0.07	1.00	0.02	-0.02	-0.01	0.02	0.00	0.02
Sulfate	0.01	-0.09	-0.15	0.02	1.00	-0.01	0.03	-0.03	-0.01	-0.02
Conductivity	0.02	-0.02	0.01	-0.02	-0.01	1.00	0.02	0.00	0.01	-0.01
Organic_carbon	0.04	0.00	0.01	-0.01	0.03	0.02	1.00	-0.01	-0.03	-0.03
Trihalomethanes	0.00	-0.01	-0.01	0.02	-0.03	0.00	-0.01	1.00	-0.02	0.01
Turbidity	-0.04	-0.01	0.02	0.00	-0.01	0.01	-0.03	-0.02	1.00	0.00
Potability	-0.00	-0.01	0.03	0.02	-0.02	-0.01	-0.03	0.01	0.00	1.00

Figure 5. Correlation heat map of the features

The dataset has class imbalance, with 1,998 records labelled as “Not Potable” and 1,278 records labelled “Potable”. This imbalance is an important consideration for designing machine learning algorithms later, as it may necessitate applying class-balancing techniques. To get a better idea about the features and their effect on potability, boxplots were generated to detect outliers, as shown in **Figure 6** to **Figure 9**.

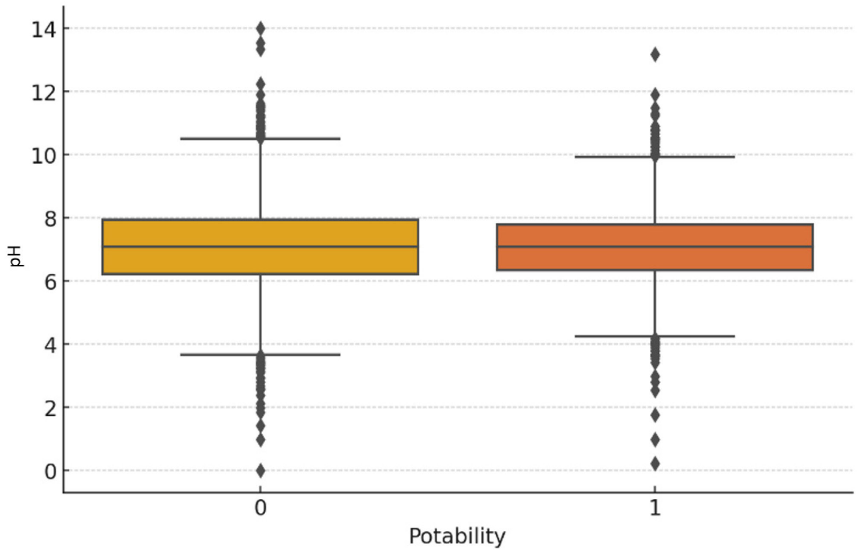


Figure 6. Boxplot for pH by potability

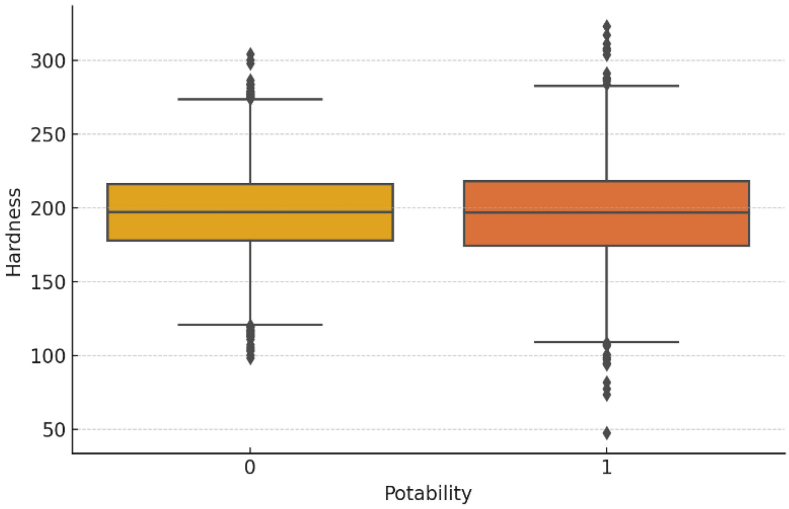


Figure 7. Boxplot for hardness [mg/L] by potability

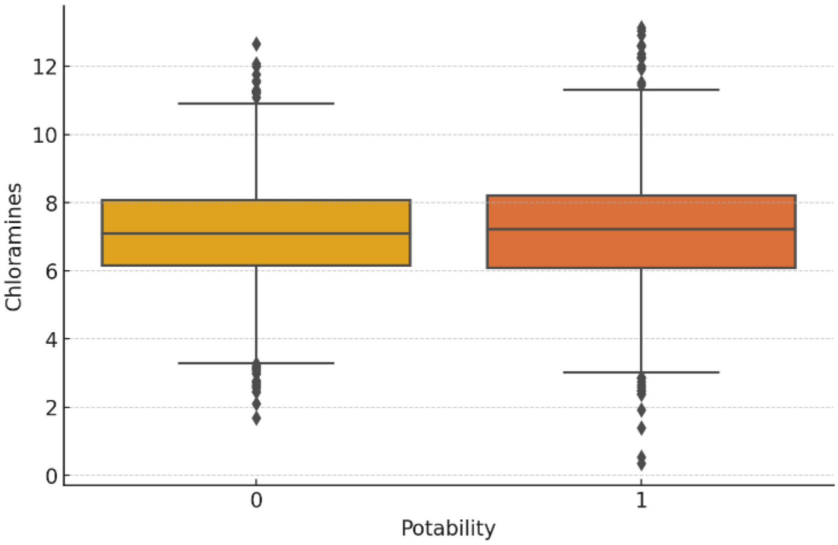


Figure 8. Boxplot for chloramines [ppm] by potability

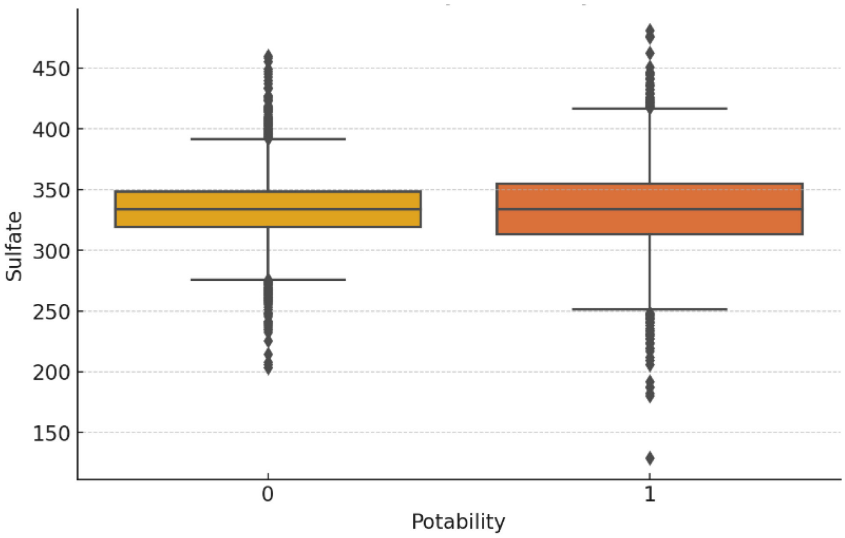


Figure 9. Boxplot for sulfate [ppm] by potability

From the above figures, it can be seen that for several features, the medians and distributions differ between potable and non-potable water, but often with overlap. Features such as pH, chloramines, sulfate, and trihalomethanes show visible differences.

Feature Importance Analysis and Dimensionality Reduction

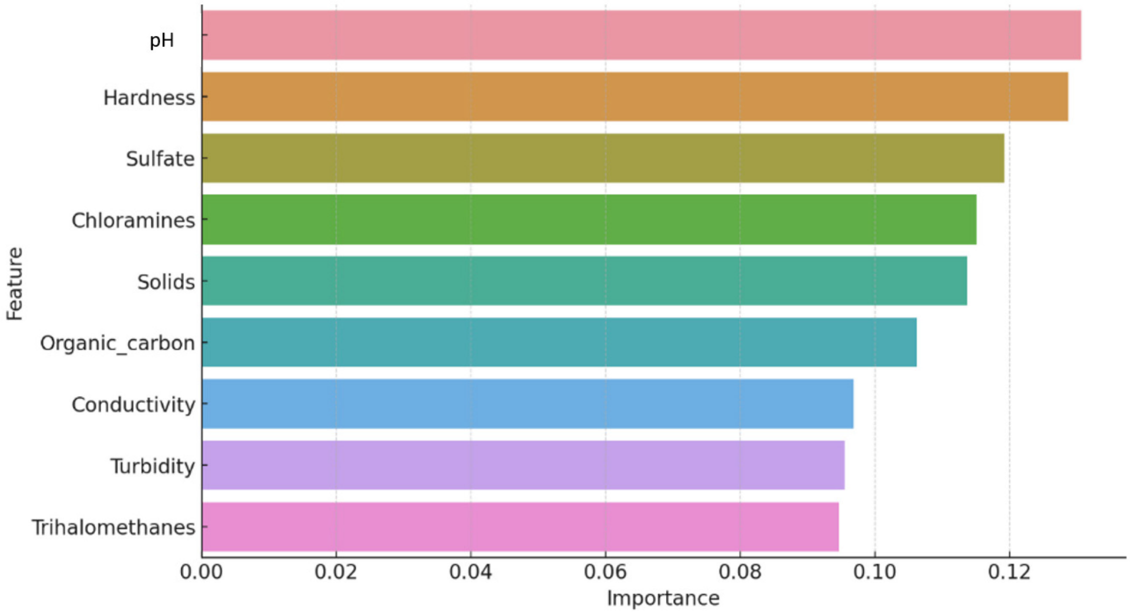


Figure 10. Feature importance analysis result

Feature importance analysis was carried out using the Random Forest Gini Importance (also known as Mean Decrease in Impurity), which measures how each feature decreases impurity in classification trees [31]. It is a default and most commonly used technique in the scikit-learn library. The method calculates the total reduction in the impurity that each feature contributes across all decision trees in the Random Forest. Figure 10 shows the result of the feature importance analysis.

The features above the importance median are pH, hardness, sulfate, chloramines, and solids. Accordingly, these five features are identified as the most relevant for predicting water potability. A refined version of the dataset was generated, which consists of the cleaned data (without any missing records) with the five features along with the potability label (0, 1).

Performance Metrics

One of the most commonly used metrics for classification algorithm performance assessment is accuracy, which, at the same time, is what is often reported [32]. However, it sometimes proves to be a deceptive and risky metric, especially with imbalanced datasets, where a class has more samples for one label than the other. In the dataset used in this work, there are 1,998 entries for the “Not Potable” class and 1,278 for the “Potable” class. The “Not Potable” class is therefore the majority class, whereas “Potable” is the minority class. If left unaddressed, machine learning algorithms tend to be biased towards predicting the majority class with deceptively high overall accuracy but poor performance when detecting the minority class [33]. Moreover, a model may have high accuracy because it is predicting the primary class; however, this does not necessarily indicate that the model has learned the true patterns within the dataset. Thus, it is vital to use other metrics such as the F1-score and AUC. The F1-score is the harmonic mean of precision (the proportion of true positive results among the total positive results) and recall (the proportion of total positives that were identified), which in turn indicates that the model is performing well in avoiding false positives and false negatives [32]. For example, while the model can accurately classify 80 safe samples out of 100 but inaccurately labels 40 unsafe samples as safe, the precision and recall will differ, and the F1-score will provide a balanced combined measure.

This approach is crucial when the cost of misclassification is high. The AUC evaluates the ability of the model to distinguish between classes for any given classification threshold. An AUC of 0.5 would indicate random guessing, while an AUC of 1.0 would indicate that the potable samples and non-potable samples are completely distinguishable. For instance, if the Random Forest algorithm has an AUC of 0.90, that means it will be accurate in 90% of the cases in ranking a randomly chosen potable sample over a non-potable sample. Accordingly, the F1 score and AUC will be considered (besides accuracy, precision, and recall) because they provide a more in-depth and reliable picture of the algorithm's real-world performance than accuracy alone.

Model Selection and Justification

In this study, four classification models were selected for their unique features in binary classification problems with tabular real-world datasets such as the water potability dataset. RF, LR, XGBoost, and a DL model, which is the MLP, provide a balance between traditional machine learning tools, advanced ensemble models, and neural network-based approaches. Below is a more detailed justification, introduced with a summary shown in **Table 1**. Further, **Table 2** provides the key parameters for each of the four models that were implemented using Python.

Table 1. Justification of model selection

Model	Justification
Random Forest	Handles nonlinear patterns, provides built-in feature importance
Logistic Regression	Serves as a baseline due to its simplicity and interpretability
XGBoost	High accuracy and scalability, robust to noise and imbalance
Deep Learning MLP	Tests how well a neural net generalises in this domain

- RF Classifier: Random Forest was chosen for its robustness, interpretability, and performance with structured data. As an ensemble of decision trees, it reduces overfitting by averaging out predictions from many trees, which makes it very useful for complex nonlinear interactions in data. Also, its structure includes a feature importance analysis, which is a valuable tool for identifying the main water quality parameters.
- LR Classifier: Logistic Regression is a simple and interpretable baseline model. It is commonly used in the field of binary classification problems, and does very well when the relationship between input features and the target variable is almost linear. Also, it serves as a performance benchmark to which more complex models can be compared.
- XGBoost Classifier: Extreme Gradient Boosting was selected due to its outstanding performance and speed when processing structured datasets. Its features, such as regularisation and parallel computation, make it more advanced than typical boosting algorithms. As it is known for its ability to model nonlinearity and complex variable interactions, which makes it a top-performing algorithm in real-world applications, it is a key player in this study.
- Deep Learning MLP: A feed-forward neural network is a large-scale learning model that can learn complex high-level abstractions from the data. Though it does not always perform best on small to medium-sized tabular datasets, Deep Learning can outperform other models when scaled up in terms of data input and tuning. It also serves as a tool to see how well a neural architecture generalises, which is in contrast to tree-based models and linear classifiers.

Table 2. Parameters of models in Python, with open-source libraries scikit-learn, xgboost, and keras

Model	Library Used	Key Parameters	Role/Description
Random Forest	scikit-learn	n_estimators = 200	Number of decision trees. More trees reduce variance and improve stability.
		max_depth = 10	Maximum depth of each tree. Prevents overfitting by limiting complexity.
		min_samples_split = 5	Minimum samples required to split a node. Controls model generalisation.
		min_samples_leaf = 2	Minimum samples required at a leaf node. Reduces overfitting on noise.
Logistic Regression	scikit-learn	solver='liblinear'	Optimisation algorithm suitable for small/medium datasets and binary classification.
		Penalty = 'l2'	Regularisation type. Prevents overfitting by penalising large coefficients.
		C = 1.0	Inverse of regularisation strength. Balances bias and variance.
XGBoost	xgboost	n_estimators = 300	Number of boosting rounds. More rounds generally improve performance but risk overfitting.
		learning_rate = 0.05	Step size shrinkage. Smaller values make learning more robust.
		max_depth = 6	Depth of individual trees. Balances complexity and generalisation.
		Subsample = 0.8	Fraction of training samples used per tree. Introduces randomness for robustness.
		colsample_bytree = 0.8	Fraction of features sampled per tree. Reduces correlation among trees.
Deep Learning MLP	TensorFlow / keras	Dense (64, relu) → Dense (32, relu) → Dense (1, sigmoid)	Neural network layers: hidden layers with ReLU activation capture nonlinearity. Final sigmoid outputs probability.
		optimizer=Adam (lr=0.001)	Adaptive optimiser controlling weight updates.
		loss=binary_crossentropy	Loss function for binary classification tasks.
		epochs = 50, batch_size = 32	Training settings. Define how long and with what batch size the model trains.

Class Imbalance Handling

As noted, the dataset has a class imbalance – an issue that must be addressed. One of the commonly used methods in this regard is SMOTE (Synthetic Minority Over-sampling Technique) [34], [35]. It generates artificial samples of the minority class by interpolating existing samples in feature space to allow the classifier to learn the decision boundary in a better way. SMOTE was applied to the training set only to avoid the data leakage problem.

Hyperparameter Tuning

The hyperparameter tuning method is used to find the best set of parameters for ML models without modifying the algorithm itself. It occurs before training. It optimises model performance in such a way that it enhances accuracy and generalisation (performance on new data), reduces overfitting or underfitting, and increases training efficiency (speed, memory usage).

Hyperparameter tuning was also conducted for the four models using GridSearchCV and 5-fold stratified cross-validation. The method tried various combinations of parameters to identify the optimal configuration according to the F1-score, improving the generalisation ability and precision of the used models.

External Validation and Deployment

To assess the external validity of the proposed approach, the findings of this work will be compared with the results in existing literature. For example, the use of ensemble models (RF and XGBoost) will be compared with the performance of these models shown in references such as [10], [18], [24], while the performance of SMOTE in imbalanced data will be assessed by studies such as [34], [35]. Although the dataset differs from other works in geography or water source, performance trends can provide a good understanding and validation for the methodological choices of this research.

After assessing the performance of each model, the best model will be exported (using joblib) and integrated into a user-friendly GUI application to be built with Streamlit. The application allows users to enter water characteristics and receive real-time predictions on potability (i.e., safe or unsafe to drink). More details will be provided in the next section.

SIMULATION RESULTS AND DISCUSSIONS

This section provides the results of the four classification models that were used to predict water potability based on the five selected features. The performance of each of the four models was evaluated using a stratified train-test split (80/20) and five performance metrics. All simulations were performed using the Python programming language.

Initial Performance Results of the Models

The first experiment, conducted on the refined dataset after handling missing values and selecting only the five top features, is to generate a performance comparison of the four models, namely, RF, LR, XGBoost, and DL MLP. It should be noted that the dataset used in this experiment still suffers from class imbalance.

After running the four models, performance metrics were calculated based on the obtained results. **Table 3** shows a summary of the metric values for the four models.

Table 3. Initial model performance comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.6714	0.6339	0.3708	0.4679	0.6837
Logistic Regression	0.6104	0	0	0	0.5234
XGBoost	0.6673	0.6111	0.4021	0.485	0.6636
Deep Learning MLP	0.6104	0	0	0	0.5

Table 2 above highlights the superior performance of RF and XGBoost, where they both outperform LR and DL in terms of F1-score and AUC. The poor performance of the LR and the DL neural network model is likely due to class imbalance and a lack of model complexity or tuning. **Figure 11** shows the F1-score results, while **Figure 12** shows the AUC score results of the four models.

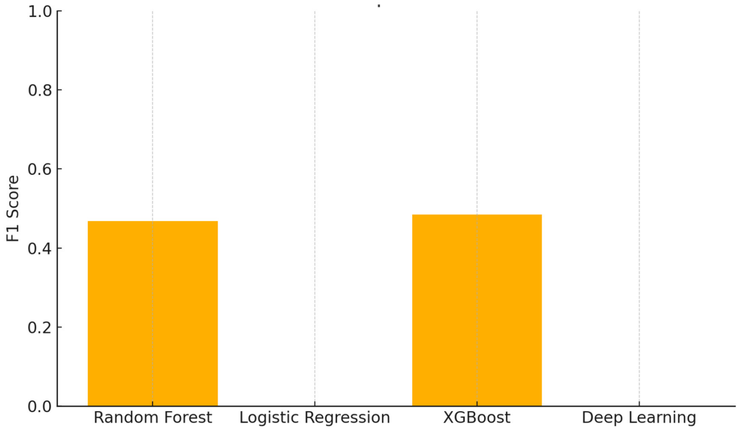


Figure 11. F1-Score Results of the Four Models

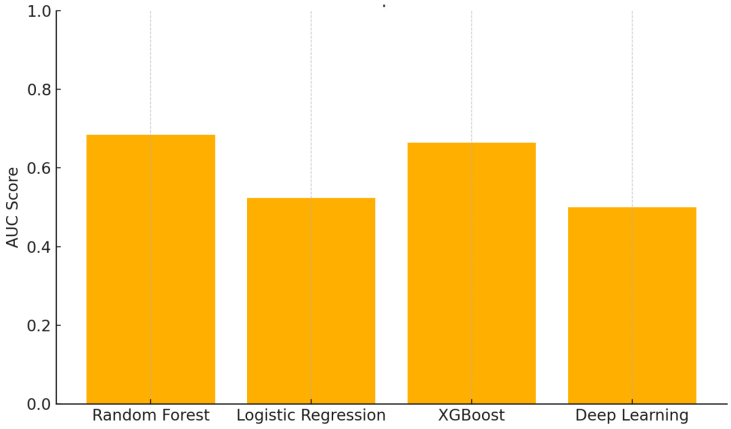


Figure 12. AUC score results of the four models

From **Figure 11**, XGBoost slightly outperformed the RF model, while LR and DL have shown poor F1 scores due to class imbalance or underfitting problems. From **Figure 12**, it can be seen that the RF model has shown the best AUC (which represents the discriminative power), followed closely by XGBoost. LR and DL performed almost randomly (since AUC ≈ 0.5).

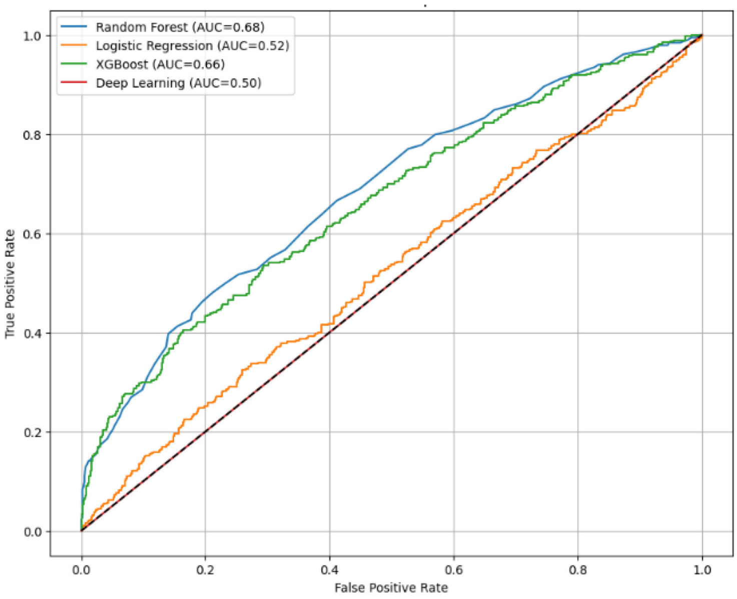


Figure 13. ROC curve comparison for the four models

Figure 13 highlights the ROC curve comparison for the four tested models, which confirms the findings stated above about the four models. Based on the metrics and considering the importance of correctly classifying potable and non-potable water, Random Forest can be considered the best-performing model for this task, closely followed by XGBoost. Random Forest has the highest AUC (0.684) and a decent F1-score (0.468), while XGBoost has a slightly lower AUC (0.664) but a slightly better F1-score (0.485). Random Forest achieved the highest AUC score, indicating better overall class discrimination ability, and XGBoost achieved the highest F1-score, reflecting a slightly better balance between precision and recall, especially when both classes are important. Logistic Regression and Deep Learning failed completely (precision, recall, and F1-score are all zeros). This likely means the models are predicting only one class (probably “non-potable”) and not learning properly. The models still have room for improvement because the best F1-scores are below 50%. This result suggests either that the features were not fully predictive or that models need better tuning (hyperparameter optimisation) or that more complex features (e.g., interaction terms) are needed.

Performance Results of the Models After Class Balancing, Hyperparameter Tuning, and Feature Engineering

Data imbalance is a known issue in classification applications that can lead to biased models favouring the majority class while hindering performance on the minority class. Among others, SMOTE is used to address this problem and increase the number of samples in the minority class. It is a popular technique for handling class imbalance by creating synthetic examples of the minority class (in this case, “potable” water) rather than simply duplicating them.

In this work, hyperparameter tuning was implemented using GridSearchCV, which is part of the scikit-learn Python library. It is a brute-force search technique that evaluates all possible combinations of specified hyperparameters using cross-validation.

Feature engineering is the process of transforming raw data into useful features for machine learning models. These engineered features capture complex interactions that may not be obvious in the default datasets. Some of the features can be selected, and new features can be created through the transformation of features. Such a process can optimise the model’s performance and make it more efficient and accurate. Four new features, using operations such as ratios, products, and differences, were added, and the performances of the four models were analysed again by incorporating the new features in the dataset. The additional new features are provided in **Table 4**.

Table 4. The new engineered features

New features	Formula	Description
Hardness_Solids_Ratio	Hardness / Solids	The relation between mineral content and solids
Sulfate_Hardness_Ratio	Sulfate / Hardness	Relative concentration of sulfate to hardness
pH_Hardness_Product	pH × Hardness	Interaction term: acidity vs. minerals
Solids_Sulfate_Diff	Solids − Sulfate	Mass difference between solids and sulfates

After carrying out the above improvements, the models were trained and tested again. **Table 5** shows the performance metrics results for the four models. It can be seen from the table that there are noticeable improvements as a result of the SMOTE-based class balancing, hyperparameter tuning, and feature engineering (e.g., hardness/solids ratio, sulfate interactions). The F1-score and AUC for all the models have improved considerably.

Table 5. Model performance comparison after class balancing, hyperparameter tuning, and feature engineering

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.8634	0.8744	0.8442	0.8533	0.9022
Logistic Regression	0.8333	0.8536	0.8131	0.8342	0.8832
XGBoost	0.8721	0.8923	0.8435	0.8623	0.9124
Deep Learning MLP	0.8132	0.8211	0.7913	0.8012	0.8541

From the above table, it can be seen that XGBoost and Random Forest models have achieved a more balanced classification between the “safe” and “unsafe” water classes, with much improved F1-score and AUC. Logistic Regression and Deep Learning models have also improved significantly after balancing the dataset and feature engineering, but showed slightly lower recall and F1-score.

Figure 14 shows the F1-score results of the four models after the above adjustments and improvements, while **Figure 15** shows the AUC score results of the four models after the same adjustments and improvements.

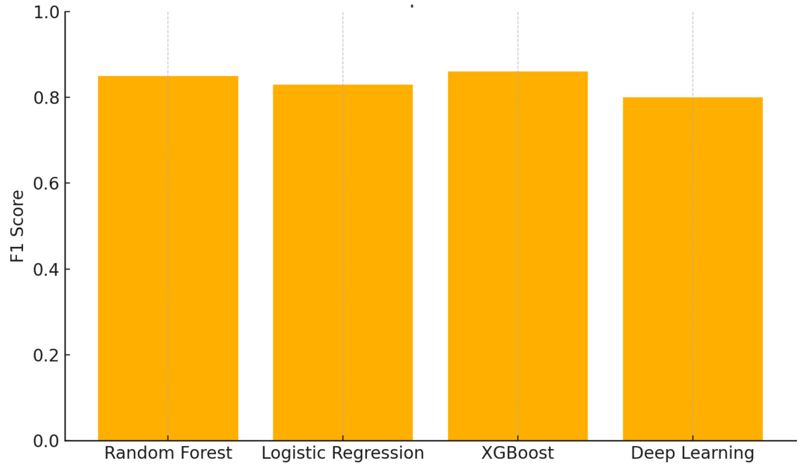


Figure 14. F1-Score Results of the Four Models after Improvements

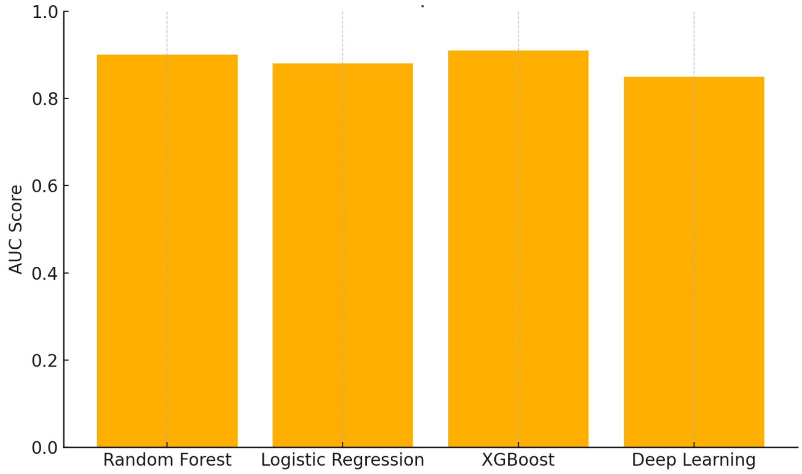
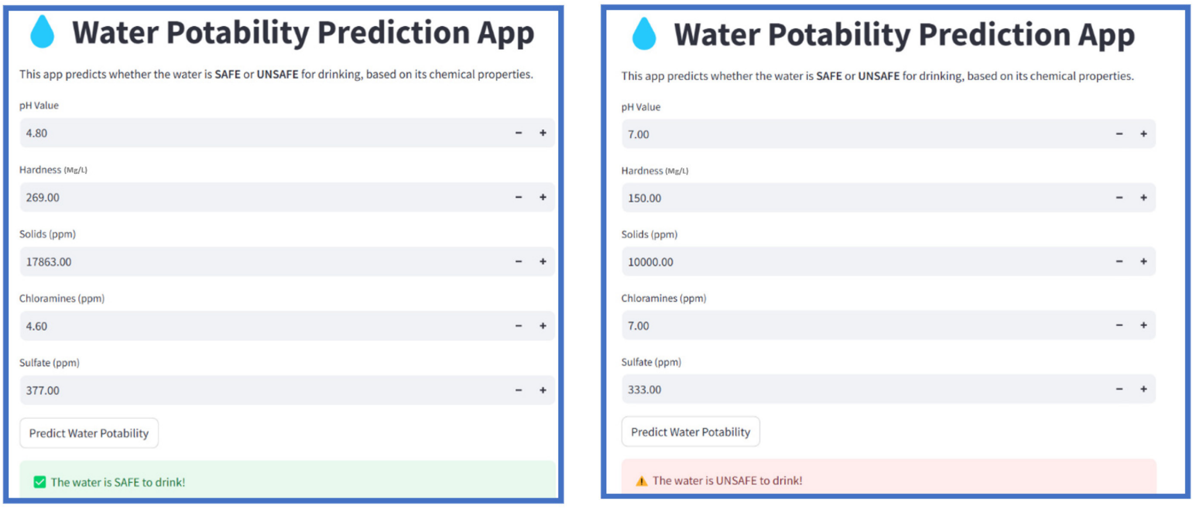


Figure 15. AUC Score Results of the Four Models after Improvements

XGBoost has shown the highest F1-score (0.86), followed by Random Forest (0.85). Logistic Regression and Deep Learning performed slightly lower. XGBoost also has the highest AUC (0.91), which confirms its superior ability to distinguish between safe and unsafe water, and Random Forest follows closely with 0.90. Logistic Regression and Deep Learning are slightly behind. Accordingly, the XGBoost model offers the best overall balance between precision and recall. Deep Learning achieved decent performance but did not outperform ensemble methods on this tabular dataset.

In summary, XGBoost is the top-performing model with the highest AUC (0.91) and an excellent F1-score (0.86). Random Forest closely follows with an AUC of 0.90 and strong classification metrics. Logistic Regression provided a strong baseline with good performance after balancing and tuning. The Deep Learning MLP was effective but less optimal than ensemble models in this context. As a result, the XGBoost model was selected for deployment due to its strong performance, interpretability, balance of accuracy, and robustness.

Lastly, an application was created using Python joblib and Streamlit. The application provides the user with a prediction of whether the water is drinking “SAFE” or “UNSAFE” after entering the values of the required chemical properties or the feature values on the GUI. The XGBoost model was saved (exported) using the joblib library. Then, the application was built using Streamlit, which is a Python-based framework for rapid app deployment. Figure 16 shows two examples of executing the application. The app can be implemented on mobile devices, PCs, or other platforms, allowing easy visual interaction with users interested in predicting water potability.



Water Potability Prediction App

This app predicts whether the water is **SAFE** or **UNSAFE** for drinking, based on its chemical properties.

pH Value: 4.80
Hardness (Mg/L): 269.00
Solids (ppm): 17863.00
Chloramines (ppm): 4.60
Sulfate (ppm): 377.00

✅ The water is SAFE to drink!

Water Potability Prediction App

This app predicts whether the water is **SAFE** or **UNSAFE** for drinking, based on its chemical properties.

pH Value: 7.00
Hardness (Mg/L): 150.00
Solids (ppm): 10000.00
Chloramines (ppm): 7.00
Sulfate (ppm): 333.00

⚠ The water is UNSAFE to drink!

Figure 16. Water Potability Prediction App – safe/unsafe water prediction outputs

Validation with Existing Literature

To ensure the external validity of this work's findings, the performances of the used models are compared to results in existing water quality prediction research. Previous relevant studies have shown that more advanced ensemble methods, such as Random Forest and gradient boosting algorithms, e.g., XGBoost, outperform baseline linear models or logistic regression. Apart from this, class balancing methods such as SMOTE have been widely verified in environmental as well as general machine learning applications to be efficient tools against imbalanced datasets. Table 6 summarises this work's findings versus those available in recent literature. Evidence presented in Table 6 verifies that the obtained results are aligned with previous studies. Specifically, ensemble models always perform well on different datasets and geographic locations, which indicates the robustness and generalisability of the used models. Also, the use of SMOTE and hyperparameter tuning conforms to the state-of-the-art practice in the literature and further justifies their necessity in improving model fairness and prediction performance in imbalanced datasets. While the dataset used in this work differs in size and origin from the ones described in the literature, the uniform patterns of performance across studies confirm the accuracy and stability of the proposed approach.

Table 6. Comparison of the proposed work findings with existing literature

Reference	Task	Models compared	Findings	Comparison with the proposed work
Ahmed *et al.* [16]	Prediction using supervised ML	RF, SVM, ANN	RF outperformed linear models; ensembles capt-ured nonlinear relations	Confirms RF superiority
Chen *et al.* [18]	Comparative analysis on surface water prediction	Multiple ML models	Tree models (RF, boosting) had higher predictive accuracy	Supports RF/XGB results
Lu & Ma [22]	Short-term water quality forecasting	Hybrid decision-tree ensembles	Ensemble models out-performed single models	Supports RF/XGB results
Bui *et al.* [24]	Water quality index prediction	Hybrid ML vs. standalone	Hybrid/ensemble ML improved	Emphasises ensemble advantage
Shams *et al.* [10]	Prediction with grid search tuning	RF, XGB, others	Boosting with tuning achieved best performance	Aligns with GridSearch-CV improvements
Fernandez *et al.* [34]	Review of SMOTE	ML models with imbalanced datasets	SMOTE improves classifier performance	Validates the use of SMOTE
Pradipta *et al.* [35]	Review of SMOTE in practice	Multiple ML models	SMOTE widely used for class imbalance	Confirms the robustness of the used balancing approach

Limitations of the Work

While this study demonstrates the applicability of ensemble and boosting techniques, class balancing, and feature engineering in water potability prediction, a few limitations exist. First, the study is limited to one publicly available dataset, which constrains the geographic and temporal range of water quality parameters under consideration. Therefore, the findings may not necessarily apply fully to water sources with other characteristics. Second, the validation method is limited to held-out test data from the same dataset. Although stratified train–test splits and cross-validation minimise overfitting, more comprehensive blind testing across external datasets from other sources or under different environmental conditions is necessary for external validation. Third, interpretability is addressed through feature importance analysis on ensemble models, but advanced interpretability tools (e.g., SHapley Additive exPlanations) were not implemented in this work. These may provide greater model decision-making insight, especially in neural networks. Finally, real-world deployment concerns such as sensor fusion, data streaming, and hardware efficiency are beyond the scope of this work, although the prototype Streamlit application demonstrates practical feasibility. These limitations are areas for future investigations. Generalisation to multiple datasets, using advanced explainability techniques, and testing in realistic operating environments would enhance the validity and generalizability of the proposed models.

Discussion

To show the impact of the improvement after class balancing, hyperparameter tuning, and feature engineering, one can compare the results shown in **Table 3** and **Table 5**. As can be seen from this comparison, all four models improved their performance significantly, with Random Forest and XGBoost improving the most. For example, Random Forest F1-score improved from 0.47 to 0.85, and XGBoost from 0.49 to 0.86, while Logistic Regression and Deep Learning MLP also saw a significant improvement from near-random performance to competitive performance.

It should be noted, though, that after feature engineering, the differences between the two top models (Random Forest and XGBoost) are very small, with XGBoost being only ~1% superior in F1-score and AUC. This result means that feature engineering is the process that gets all models to a strong and comparable performance. The marginal gain of XGBoost may justify its selection as the deployment model, but Random Forest is a close second. RF may be preferred in circumstances where simpler interpretability or reduced computational cost is desirable.

CONCLUSIONS

This study presented a comparative analysis of four prominent machine learning models: Random Forest, XGBoost, Linear Regression and Deep Learning MLP on a certain water potability dataset based on key water quality features. LR was chosen for its simplicity and interpretability, RF – for its robustness and generalisation, XGBoost – for its high-performance gradient boosting, and the Deep Learning MLP model – for its generalisation ability. This diverse selection offers meaningful insights into which algorithm is most suitable for water potability classification based on both performance metrics and practical deployment considerations. Initial experiments showed moderate performance, with RF and XGBoost outperforming the other two models. The traditional LR model performed less efficiently in terms of predictive accuracy and robustness. While it remains valuable for its simplicity and interpretability, it proved less effective in handling data with several features, imbalanced or nonlinear data.

To further enhance performance, class balancing, hyperparameter tuning, and feature engineering techniques were applied to create new features capturing the relationships between water properties. After these improvements, both RF and XGBoost achieved significantly better F1-scores and maintained competitive AUC scores. The Deep Learning MLP model has proved less efficient, which could probably indicate that it requires further tuning and potentially more data to generalise well. RF and XGBoost models, after tuning and feature engineering, offer strong and balanced performance for water potability prediction. A user-friendly application with a simple GUI was developed to provide real-time prediction of water potability.

This research shows that model selection should be guided by the specific characteristics of the dataset, performance requirements, and available computational resources. Future work could explore even more advanced features, ensemble methods, hybrid models, or collect additional datasets to optimise the performance of the models further and attain enhanced predictions.

NOMENCLATURE

Abbreviations

ANN	Artificial Neural Network
AUC	Area Under the ROC Curve
CNN	Convolutional Neural Network
DL	Deep Learning
GUI	Graphical User Interface
IoT	Internet of Things
LR	Logistic Regression

LSTM	Long Short-Term Memory network
MLP	Multilayer Perceptron (feed-forward NN)
ML	Machine Learning
RF	Random Forest
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boosting

ACKNOWLEDGMENTS

The authors appreciate the support provided by the School of Engineering and Computing, American University of Ras Al Khaimah.

REFERENCES

1. D. Sharma and A. Kansal, “Water quality analysis of River Yamuna using water quality index in the national capital territory, India (2000–2009),” *Appl Water Sci*, vol. 1, no. 3–4, pp. 147–157, Dec. 2011, https://doi.org/10.1007/s13201-011-0011-4.
2. H. Faraji and A. Shahryari, “Assessment of groundwater quality for drinking, irrigation, and industrial purposes using water quality indices and GIS technique in Gorgan aquifer,” *Desalination Water Treat*, vol. 320, p. 100821, Oct. 2024, https://doi.org/10.1016/j.dwt.2024.100821.
3. J. Halder and N. Islam, “Water Pollution and its Impact on the Human Health,” *Journal of Environment and Human*, vol. 2, no. 1, pp. 36–46, Jan. 2015, https://doi.org/10.15764/EH.2015.01005.
4. U. Ahmed, R. Mumtaz, H. Anwar, S. Mumtaz, and A. M. Qamar, “Water quality monitoring: from conventional to emerging technologies,” *Water Supply*, vol. 20, no. 1, pp. 28–45, Feb. 2020, https://doi.org/10.2166/ws.2019.144.
5. A. Babatunde, “Study on traditional water quality assessment methods,” *Assessment and Management Decisions*, vol. 1, no. 1, pp. 41–52, Jul. 2024.
6. P. Khatri, K. K. Gupta, and R. K. Gupta, “Assessment of Water Quality Parameters in Real-Time Environment,” *SN Comput Sci*, vol. 1, no. 6, p. 340, Nov. 2020, https://doi.org/10.1007/s42979-020-00368-9.
7. V. Madhavireddy and B. Koteswarrao, “Smart Water Quality Monitoring System Using Iot Technology,” *International Journal of Engineering & Technology*, vol. 7, no. 4.36, pp. 636–639, Dec. 2018, https://doi.org/10.14419/ijet.v7i4.36.24214.
8. V. Lakshmikantha, A. Hiriyannagowda, A. Manjunath, A. Patted, J. Basavaiah, and A. A. Anthony, “IoT based smart water quality monitoring system,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 181–186, Nov. 2021, https://doi.org/10.1016/j.gltp.2021.08.062.
9. B. Das and P. C. Jain, “Real-time water quality monitoring system using Internet of Things,” in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, IEEE, Jul. 2017, pp. 78–82, https://doi.org/10.1109/COMPTELIX.2017.8003942.
10. M. Y. Shams, A. M. Elshewey, E.-S. M. El-kenawy, A. Ibrahim, F. M. Talaat, and Z. Tarek, “Water quality prediction using machine learning models based on grid search method,” *Multimed Tools Appl*, vol. 83, no. 12, pp. 35307–35334, Sep. 2023, https://doi.org/10.1007/s11042-023-16737-4.
11. S. Peerzade and P. Kamat, “Enhancing water quality prediction: a machine learning approach across diverse water environments,” *Water Quality Research Journal*, vol. 60, no. 1, pp. 298–317, Feb. 2025, https://doi.org/10.2166/wqrj.2025.083.

12. S. Chatterjee, S. Sarkar, N. Dey, S. Sen, T. Goto, and N. C. Debnath, “Water quality prediction: Multi objective genetic algorithm coupled artificial neural network based approach,” in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, IEEE, Jul. 2017, pp. 963–968. https://doi.org/10.1109/INDIN.2017.8104902.
13. F. Abbas *et al.*, “Machine Learning Models for Water Quality Prediction: A Comprehensive Analysis and Uncertainty Assessment in Mirpurkhas, Sindh, Pakistan,” *Water (Basel)*, vol. 16, no. 7, p. 941, Mar. 2024, https://doi.org/10.3390/w16070941.
14. K. K, S. Krishnan, and R. Manikandan, “Water quality prediction: a data-driven approach exploiting advanced machine learning algorithms with data augmentation,” *Journal of Water and Climate Change*, vol. 15, no. 2, pp. 431–452, Feb. 2024, https://doi.org/10.2166/wcc.2023.403.
15. Y. Liu, Y. Liang, K. Ouyang, S. Liu, D. Rosenblum, and Y. Zheng, “Predicting Urban Water Quality with Ubiquitous Data - A Data-driven Approach,” *IEEE Trans Big Data*, pp. 1–1, 2020, https://doi.org/10.1109/TBDATA.2020.2972564.
16. U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, “Efficient Water Quality Prediction Using Supervised Machine Learning,” *Water (Basel)*, vol. 11, no. 11, p. 2210, Oct. 2019, https://doi.org/10.3390/w11112210.
17. J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, “Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, Nov. 2017, pp. 1058–1065, https://doi.org/10.1109/ICDMW.2017.149.
18. K. Chen *et al.*, “Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data,” *Water Res*, vol. 171, p. 115454, Mar. 2020, https://doi.org/10.1016/j.watres.2019.115454.
19. N. Mahesh, J. J. Babu, K. Nithya, and S. A. Arunmozhi, “Water quality prediction using LSTM with combined normalizer for efficient water management,” *Desalination Water Treat*, vol. 317, p. 100183, Jan. 2024, https://doi.org/10.1016/j.dwt.2024.100183.
20. Y. Wang, J. Zhou, K. Chen, Y. Wang, and L. Liu, “Water quality prediction method based on LSTM neural network,” in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, IEEE, Nov. 2017, pp. 1–5, https://doi.org/10.1109/ISKE.2017.8258814.
21. S.-S. Baek, J. Pyo, and J. A. Chun, “Prediction of Water Level and Water Quality Using a CNN-LSTM Combined Deep Learning Approach,” *Water (Basel)*, vol. 12, no. 12, p. 3399, Dec. 2020, https://doi.org/10.3390/w12123399.
22. H. Lu and X. Ma, “Hybrid decision tree-based machine learning models for short-term water quality prediction,” *Chemosphere*, vol. 249, p. 126169, Jun. 2020, https://doi.org/10.1016/j.chemosphere.2020.126169.
23. R. Barzegar, M. T. Aalami, and J. Adamowski, “Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model,” *Stochastic Environmental Research and Risk Assessment*, vol. 34, no. 2, pp. 415–433, Feb. 2020, https://doi.org/10.1007/s00477-020-01776-2.
24. D. T. Bui, K. Khosravi, J. Tiefenbacher, H. Nguyen, and N. Kazakis, “Improving prediction of water quality indices using novel hybrid machine-learning algorithms,” *Science of The Total Environment*, vol. 721, p. 137612, Jun. 2020, https://doi.org/10.1016/j.scitotenv.2020.137612.
25. M. Zhu *et al.*, “A review of the application of machine learning in water quality evaluation,” *Eco-Environment & Health*, vol. 1, no. 2, pp. 107–116, Jun. 2022, https://doi.org/10.1016/j.eehl.2022.06.001.
26. M. T.R. *et al.*, “Water quality level estimation using IoT sensors and probabilistic machine learning model,” *Hydrology Research*, vol. 55, no. 7, pp. 775–789, Jul. 2024, https://doi.org/10.2166/nh.2024.048.

27. H. Juahir *et al.*, “Spatial water quality assessment of Langat River Basin (Malaysia) using environmetric techniques,” *Environ Monit Assess*, vol. 173, no. 1–4, pp. 625–641, Feb. 2011, https://doi.org/10.1007/s10661-010-1411-x.
28. T. Zhang, J. Wu, H. Chu, J. Liu, and G. Wang, “Interpretable Machine Learning Based Quantification of the Impact of Water Quality Indicators on Groundwater Under Multiple Pollution Sources,” *Water (Basel)*, vol. 17, no. 6, p. 905, Mar. 2025, https://doi.org/10.3390/w17060905.
29. Tahsin Fuad Hasan, N. A. Kabashi, T. Saleh, M. Z. Alam, M. F. Wahab, and A. Hamid Nour, “Water Quality Monitoring Using Machine Learning And Iot: A Review,” *Chemical and Natural Resources Engineering Journal (Formally known as Biological and Natural Resources Engineering Journal)*, vol. 8, no. 2, pp. 32–54, Dec. 2024, https://doi.org/10.31436/cnrej.v8i2.100.
30. Aditya Kadiwal, “Water Potability Dataset,” https://www.kaggle.com/datasets/adityakadiwal/water-potability, [Accessed: Apr. 10, 2025].
31. R. Dunne *et al.*, “Thresholding Gini variable importance with a single-trained random forest: An empirical Bayes approach,” *Comput Struct Biotechnol J*, vol. 21, pp. 4354–4360, 2023, https://doi.org/10.1016/j.csbj.2023.08.033.
32. O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Sci Rep*, vol. 14, no. 1, p. 6086, Mar. 2024, https://doi.org/10.1038/s41598-024-56706-x.
33. N. V. Chawla, “Data Mining for Imbalanced Datasets: An Overview,” in *Data Mining and* Knowledge *Discovery Handbook*, New York: Springer-Verlag, pp. 853–867, https://doi.org/10.1007/0-387-25465-X_40.
34. A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, Apr. 2018, https://doi.org/10.1613/jair.1.11192.
35. G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, “SMOTE for Handling Imbalanced Data Problem : A Review,” in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, IEEE, Nov. 2021, pp. 1–8, https://doi.org/10.1109/ICIC54025.2021.9632912.

Paper submitted: 04.07.2025
Paper revised: 07.09.2025
Paper accepted: 17.10.2025