Article | Received 17 March 2025; Revised 6 July 2025; Accepted 31 July 2025; Published 13 August 2025
https://doi.org/10.55092/rl20250006

Combination of decision making and machine learning for improvement of robot learning for water analysis

Taraneh Javanbakht[1,*], Arbnor Pajaziti[2] and Shaban Buza[2]

[1] Department of Mechanical Engineering, École de Technologie Supérieure, Montreal, Quebec H3C 1K3, Canada

[2] Faculty of Mechanical Engineering, University of Prishtina, Prishtina 10000, Kosovo

* Correspondence author; E-mail: taraneh.javanbakht@etsmtl.ca.

Highlights:

- This paper presents the first combination of decision making and machine learning for water analysis for improvement of robot learning.
- It includes the analysis results of water samples with the development of a robotic system.
- We present the current challenges in research and discuss the perspective of this work for water analysis on Earth and other planets with the developed robotic system.

Abstract: This study presents a new investigation on the improvement of robot learning for water analysis with the combination of decision making and machine learning (ML) processes for a robotic system. The aim of the study was to perform simulations for the distinction of drinking and undrinkable water for further implementation in a robot. The decision-making process was performed with the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The data analysis with ML was done by using Microsoft Visual Studio. The TOPSIS analysis showed that the candidates having high values of profit criteria and low values of cost criteria had a better rank. The same result was obtained in the analysis of the physicochemical properties of water as well as its ingredients. The differences in the closeness coefficient values of the best and the worst candidates were 35%, and 45% in the first and second series of analyses, respectively. The ML simulation showed that using the modified code could improve the learning accuracy to 69%, which improved to 73% after using the Synthetic Minority Over-sampling Technique (SMOTE) for class balancing and applying GridSearchCV to tune the hyperparameters. The electronic components of a robotic system and the remote control of its prototype for further application of the current work have also been presented. The obtained results could be used for the implementation of the combined software in a robotic system for water analysis.

Keywords: decision-making process; TOPSIS; machine learning; robotics; water analysis

1. Introduction

Robot learning is an important ability required for the development of robots that can make them learn tasks and perform activities efficiently. This procedure is based on the application of leveraging skills and training of robots [1,2]. Robot learning has been investigated for the development of robots' different tasks such as object manipulation, item cleaning and interactive or multi-task learning [3–5]. However, it has not been performed for water analysis considering the combination of decision-making and machine learning (ML) processes. The novelty of the current work lies in the combination of a decision-making process and ML for the analysis of water for a further application in a robotic system.

Water is an important vital source for human life. Ions and organic contaminants affect the quality of water. These toxic materials have caused health and environmental problems worldwide [6]. Heavy metals that are persistent in the environment are among the most toxic water pollutants [7]. Wastewater is released from the industrial process contains many heavy metals and can pollute surface water [8,9]. The application of a decision-making process using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) has led to the analysis of the physicochemical properties of water for a further application with the new generation of robots [10,11]. This method has been widely used for categories distinction, decision making and classification. The output of TOPSIS is obtained according to the candidates' similarity to the ideal solution and the ranking is performed using the values of their closeness coefficient [12–16]. Previous investigations revealed the efficiency of TOPSIS for the analysis of drinking and undrinkable water for robotics on Earth and other planets. These research works showed that water samples could be classified according to their physicochemical and contaminant concentrations with this method. The unmodified and modified TOPSIS were used for the analysis of water samples according to their properties and concentrations of their ions and organic contaminants for the analysis of drinkable and undrinkable water on Earth and other planets with a further application of robotics [10,11].

Decision making and learning are studied separately in most research works. However, these processes are related to each other. In human cognition, the categorization of concepts, decision making and learning are the three fundamental cognitive processes. Decision making based on categorization is a preliminary step for learning. These processes have been investigated in artificial intelligence (AI), previously [17–19].

Decision making and ML as different processes of AI can be performed using the TOPSIS algorithm and Microsoft Visual Studio software, respectively. The robotic system is an autonomous machine designed and developed for the applications of simulation software [20–22].

Python is an appropriate, efficient and flexible high-level programming language in which several decision-making algorithm and ML codes have been implemented, previously. Programming in Python is simple and easy to understand. Moreover, its open-source nature is beneficial, which has given a wide range of applications in AI [23,24].

The processes of AI including categorization, decision making and ML to be considered for the code in Python are shown in Figure 1. As shown in the figure, categorization is essential for decision making, which in turn is the preliminary step for machine learning. In other terms, ML is related to decision making, which is based on categorization.

Recently, several studies that could have great impacts on robotics have been performed on mobile robot localization and its optimization [25,26], deep learning-based big data fusion for ocean water monitoring [27], optimizing underwater connectivity through multi-attribute decision-making wireless communication and development of marine sensor networks [28] and CR-NBEER: cooperative-relay neighboring-based energy efficient routing protocol for marine underwater sensor networks [29,30]. These works could create new perspectives for the development of robotic systems for water treatment.

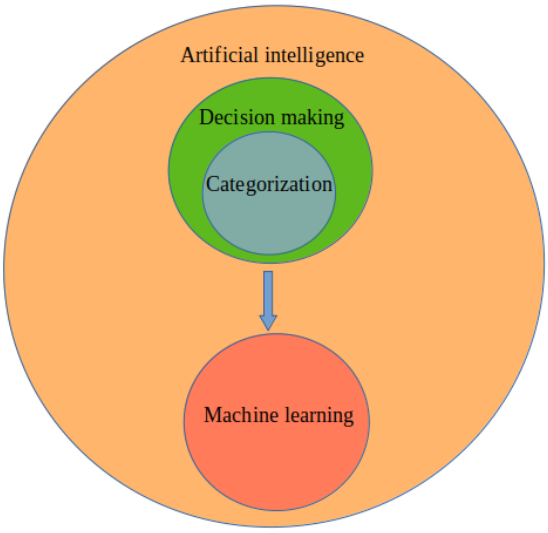

Figure 1. Categorization, decision making and ML as different processes of AI.

The objective of this work was to improve robot learning for water analysis. This study aimed to enhance water analysis by integrating decision-making (TOPSIS) and ML (Random Forest) into a unified framework for robotics. To the authors' knowledge, the analysis of ions and organic contaminants of water using the combination of decision making and ML for the improvement of robot learning has not been performed for a robotic system, yet. This work will help the robot analyze and distinguish drinking and undrinkable water.

2. Methods

The World Health Organization (WHO) guideline

The WHO guideline for naturally occurring chemicals was used for the consideration of the standard concentrations of ions in drinking water as described previously [11].

2.1. Dataset preprocessing

Dataset preprocessing was performed before analyzing the data with TOPSIS and ML. As some data was lacking in the dataset, the data table was completed before use. This step was required to improve the results.

2.2. TOPSIS method

The analysis of ions and organic contaminants and physicochemical properties of water samples such as their total dissolved solids, hardness, electrical conductivity and cost was performed with TOPSIS, previously [10,11]. The TOPSIS code in Python is available on the Glitchfix page on the GitHub website [10]. The total weight values of each candidate's characteristics, which were those of the profit and cost criteria for each analysis, were set to 1.0.

The investigated water contaminants were calcium, magnesium, boron, copper, nitrite, sodium, chloride, 1,2-dichlorobenzene, monochloramine, dichloroisocyanurate and toluene. The analyses were performed according to the model of the tree, a developed model designed for modeling the categories of concepts and the demarcator theory, a new cognitive theory of categorization, which considers the dynamic characteristics change of the members of categories in human cognition [10,11,17]. In the model of the tree, each category was considered as a tree on which the category members and their characteristics were added. This model with the combination of logic helped determine why, when and how human beings would confuse the categories. The demarcator theory considered three types of characteristics for the category members, which were a strong demarcator, weak demarcators and marginal demarcators, which helped strongly the distinction of categories, helped weakly their distinction, and did not help for it, respectively. The modified TOPSIS was used according to the model of the tree and demarcator theory, in which the algorithm modification allowed the non-consideration of the size and color of the category members (marginal demarcators) that did not have any impact on their distinction [17].

In the current work, several series of analysis were performed. In both analyses, the mean values of the triangular fuzzy data were used in the evaluation matrices. The criteria were grouped into a benefit indicator such as optimal pH and cost indicators such as turbidity, hardness and dissolved oxygen. The process involved constructing a normalized decision matrix, followed by a weighted normalized matrix. Subsequently, the Positive Ideal Solution and the Negative Ideal Solution were identified. The closeness coefficient for each sample was calculated using the formula:

$$CC_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (1)$$

where D_i^- and D_i^+ represent the distances to the negative and positive ideal solutions, respectively. Samples having the closeness coefficient with higher values were interpreted as more likely to be drinkable.

In the first series of analysis, 10 water samples were considered as candidates and their physicochemical properties such as pH, turbidity, hardness and dissolved oxygen were analyzed. The first, second, third to tenth candidates are indicated as C_1, C_2, C_3 to C_{10}, respectively. The candidates having high values of profit criteria and low values of cost criteria have a better rank. The pH value should not be low in drinking water because the acidic pH is not appropriate in drinking water samples. Therefore, pH was considered as a profit criterion. It is worth noting that very high pH values are not appropriate for drinking water and the neutral pH value is appropriate for water consumption. High levels of dissolved oxygen in drinking water do not directly cause health risk to humans, but they can affect water quality. Moreover, it can lead to increased corrosion in pipes and may impact the taste and odor of drinking water. High amount of turbidity and hardness in water could also affect its quality. Therefore, they were considered as cost criteria. The ingredient concentration of the first, second, third to tenth candidates are

indicated in the second, third, fourth to eleventh rows in the table representing the evaluation matrix. In the second series of analysis, 4 water samples were analyzed considering the concentrations of their 12 ions and 4 organic contaminants such as calcium, magnesium, boron, copper, cadmium, chromium, arsenic, nickel, lead, nitrite, sodium, chloride, 1,2-dichlorobenzene, monochloramine, dichloroisocyanurate and toluene. The first, second, third and fourth candidates are indicated as C_1, C_2, C_3 and C_4, respectively. Calcium and magnesium were considered as profit criteria, whereas other ingredients were considered as cost criteria. The ingredient concentrations of the first, second, third and fourth candidates are indicated in the second, third, fourth and fifth rows in the table representing the evaluation matrix. Another series of analysis was also performed in combination with ML.

2.3. Machine learning for water analysis

Machine learning (ML) was done by using Microsoft Visual Studio Code of Python. The dataset including the information on more than 3200 water samples used for the analyses was available in the dataset section of the Kaggle website; "Water Quality and Potability Dataset [31]." Microsoft Visual Studio was used as the integrated development environment (IDE) for implementing and testing the Python-based TOPSIS and ML models due to their robust debugging tools, ease of code management, and compatibility with Python libraries.

In the current work, we used the Random Forest Classifier, a supervised ML algorithm, for water analysis. This algorithm was employed to classify water samples as drinkable or undrinkable based on physicochemical properties.

In the first step, a dataset was constructed comprising various water samples, each characterized by a range of physicochemical properties such as pH, turbidity, electrical conductivity and dissolved oxygen level. These features were used to classify the samples as either "Drinkable" or "Undrinkable," based on established World Health Organization (WHO) thresholds.

In the second step, the TOPSIS method was applied to assess and rank the water samples according to multiple evaluation criteria.

In the third step, the closeness coefficients and rankings derived from the TOPSIS analysis were used to refine the class labels of the dataset. This approach was particularly useful for handling ambiguous or borderline cases. By incorporating expert-based multi-criteria decision-making into the labelling process, the dataset was enhanced with more reliable supervision prior to training the ML model.

In the fourth step, a Random Forest classifier was trained using the refined dataset, with physicochemical attributes as inputs and the adjusted labels as targets. To address the imbalance in class distribution, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. Additionally, hyperparameter tuning was performed using GridSearchCV to optimize the model's configuration, including parameters such as the number of trees and maximum depth. Cross-validation was also employed to ensure model generalizability and reduce overfitting risks. We chose SMOTE over random oversampling and under sampling because it generates synthetic minority class samples rather than duplicating existing ones or removing majority instances. This helps reduce overfitting and preserves information from both classes. In our tests, SMOTE provided better classification performance and generalization, making it the most suitable choice for our imbalanced dataset.

In the fifth step, the performance of the ML model was evaluated. The baseline model, trained without the benefit of TOPSIS-based label refinement, achieved an accuracy of 69%. After integrating

the TOPSIS-enhanced labelling along with class balancing and hyperparameter optimization, the model's accuracy increased to 73%. This improvement highlights the added value of combining decision-making techniques with ML, particularly in domains involving noisy or limited data, such as robotic water analysis.

3. Results and discussion

3.1. TOPSIS analysis

Several series of analysis were performed with the TOPSIS algorithm. The first series of analysis was performed on the physicochemical properties of 10 candidates (water samples) with pH values close to 7.

Table 1 shows the matrix of the means values of fuzzy data used for classification of water samples with the TOPSIS method.

Table 1. Mean values of fuzzy data for water samples.

Candidates/Criteria	pH	Turbidity	Hardness	Dissolved oxygen
C_1	6.936350	0.102922	202.963224	10.467904
C_2	8.376786	4.849549	84.873465	6.534717
C_3	7.829985	4.162213	123.036162	5.585464
C_4	7.496646	1.061696	141.590461	13.539970
C_5	6.390047	0.909125	164.017496	13.690688
C_6	6.389986	0.917023	246.293990	12.275576
C_7	6.145209	1.521211	99.918446	7.741524
C_8	8.165440	2.623782	178.558610	5.879049
C_9	7.502788	2.159725	198.103642	11.158097
C_{10}	7.770181	1.456146	61.612603	8.961372

Table 2 shows the matrix of the weight values applied for each water sample.

Table 2. Weight values applied for each water sample.

Alternatives/Values	pH	Turbidity	Hardness	Dissolved oxygen
C_1–C_{10}	0.25	0.25	0.25	0.25

Table 3 represents the criteria matrix for water samples.

Table 3. The criteria matrix for water samples.

Alternatives/Values	pH	Turbidity	Hardness	Dissolved oxygen
C_1–C_{10}	True	False	False	False

Table 4 shows the distances from the ideal solutions, the similarity coefficients, and the ranking of candidates.

Mean values of fuzzy data concerning pH, turbidity, hardness and dissolved oxygen have been provided for 10 water samples in Table 1. As shown in Table 2, the weight values of pH, turbidity, hardness and dissolved oxygen for water samples have been set to 0.25 and their total value has been set to 1.0. As shown in Table 3, pH has been considered as a profit criterion as low pH value would not be appropriate for drinking water, whereas turbidity, hardness and dissolved oxygen which high values would be inappropriate in samples have been considered as cost criteria. The ranking of water samples has been obtained according to their similarity coefficients as shown in Table 4. The results presented in Tables 1 to 4 show the impact of selecting criteria and their fuzzy values on the candidates' ranks.

Table 4. The distances from the ideal solutions, the similarity coefficients and the ranking of candidates.

Candidates	di*	di^-	CC_i	Ranking
C_1	0.08124835	0.1574548	0.65962599	10
C_2	0.15428165	0.1006233	0.39474831	7
C_3	0.13500519	0.09285198	0.40750079	1
C_4	0.08094839	0.13389381	0.62321932	4
C_5	0.08824742	0.13391575	0.60278106	5
C_6	0.11074069	0.12783012	0.53581626	6
C_7	0.05773852	0.13799795	0.7050191	8
C_8	0.10004669	0.10287733	0.50697462	9
C_9	0.10489376	0.09361219	0.47158378	3
C_{10}	0.05168493	0.14863701	0.74199066	2

Figure 2 shows the parameters of table 4 for candidates.

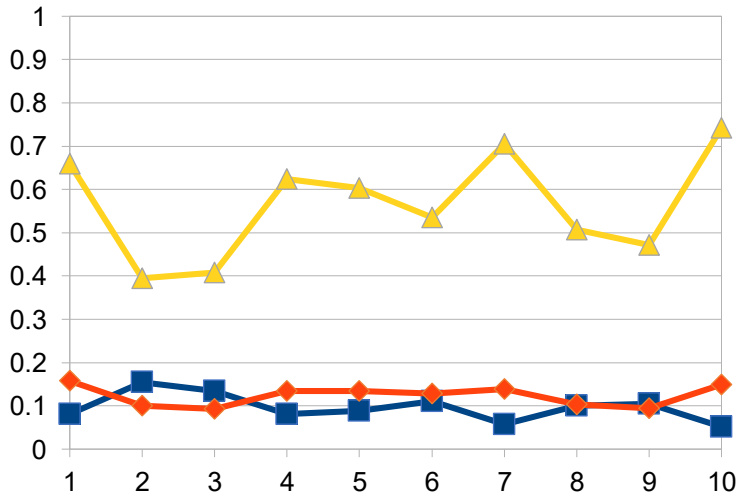

Figure 2. The candidates' best alternative is represented in blue, the worst alternative in red and similarity coefficients in yellow. The spots from left to right represent candidates 1 to 10, respectively.

The results presented in Table 4 and Figure 2 showed that low values of cost criteria and high value of profit criterion could have a significant impact on the candidates' ranks. The second and the tenth candidates were ranked as the worst and the best candidates, respectively. The difference in their

closeness coefficient values was 35%. Other candidates had other positions in the ranking according to the values of their profit and cost criteria.

The second series of analysis was performed on the contaminant concentrations of 4 candidates.

Table A1 shows the matrix of the means values of fuzzy data used for classification of water samples with the TOPSIS method, table A2 shows the matrix of the weight values and table A3 represents the criteria matrix for water samples (see the Appendix).

Table A4 shows the distances from the best and worst alternatives (di^* and di^-), the similarity coefficients (CC_i), and the ranking of candidates (see the Appendix).

Figure 3 shows the best and worst alternatives (di^* and di^-) and the similarity coefficients (CC_i) of candidates.

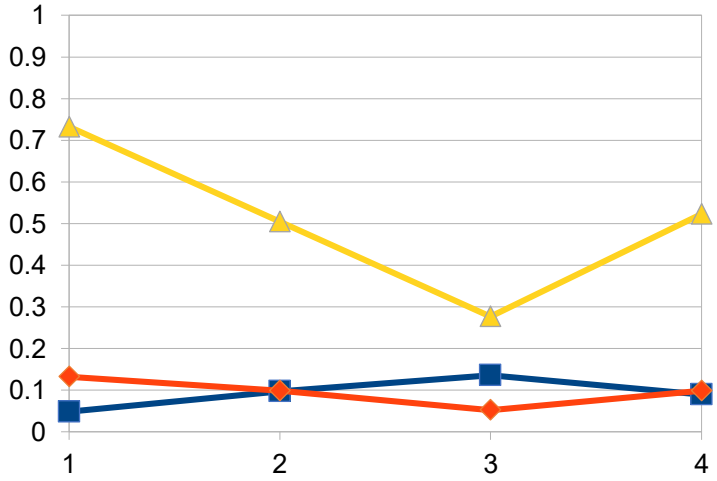

Figure 3. The candidates' best alternative is represented in blue, the worst alternative in red and similarity coefficients in yellow. The spots from left to right represent candidates 1 to 4, respectively.

The results presented in Table A4 in Appendix and Figure 3 showed that low concentrations of water contaminants and high concentrations of appropriate ions could have a significant impact on the candidates' ranks. The third and the first candidates were ranked as the worst and the best candidates, respectively. The difference in their closeness coefficient values was 45%. Other candidates had other positions in the ranking according to the values of their profit and cost criteria.

In the first series of analysis, the best ranked and the worst ranked water samples were differentiated with a 35% difference in their closeness coefficient values whereas in the second series of analysis, this reached 45%, which revealed the appropriate candidates' ranking with TOPSIS.

3.2. ML results

The ML results obtained in the current work were as below.

Confusion Matrix:

363	49
159	85

Table A5 represents the classification report and table A6 shows the characteristic data (see the Appendix).

The data in the table above used for analysis with TOPSIS correspond to the samples 1955, 2768, 3162, 2630 and 3150 in the dataset, respectively. The potability values of 1 and 0 are attributed to the drinkable and undrinkable water samples, respectively.

The samples 1955, 2768, *etc.* were selected based on their TOPSIS scores—specifically, they were among the highest-ranked samples according to the TOPSIS method. However, whether they were representative or edge cases depended on a few factors:

These Samples have been selected by using the following methods:

(1) TOPSIS Ranking

The dataset was ranked by TOPSIS Score, which evaluates how close each sample is to the "ideal" water quality conditions. The highest-ranking samples were displayed at the top of df_sorted.head().

(2) ML Classification

After sorting by TOPSIS Score, the classifier's predictions (Model_Prediction) were compared with the actual Potability labels. The selected samples likely represent cases where the model and TOPSIS agreed on classification.

Regarding the question, are these samples representative or edge cases, in most of the dataset has similar feature distributions, these samples may be representative. If these samples have extreme values (e.g., very high pH, solids, or sulfate levels), they may be edge cases that were ranked highly by TOPSIS but may not represent typical water samples.

The model achieves 68.3% accuracy, which is reasonable but could be improved. The precision for drinkable water (1) is 63%, meaning the model is slightly weaker at correctly classifying safe water. The recall for drinkable water (1) is 35%, which suggests many drinkable samples are being misclassified as not drinkable. The model favours "Not Drinkable" predictions, which could be due to class imbalance.

The improvement of the ML results could be obtained with the following procedures. 1) Feature Engineering: Refining water quality parameters using domain knowledge. 2) Hyperparameter Tuning: Optimizing the Random Forest model. 3) Balancing the Classes: Applying Synthetic Minority Over-sampling Technique (SMOTE) or cost-sensitive learning.

The recall for drinkable water being only 35% suggests that the model does not identify all the true positives (*i.e.*, all the cases where the water is drinkable). In other words, only 35% of the actual drinkable water samples are correctly identified by the model. The reason why this could happen and the risks it poses in real-world deployment are important to consider. Improving the recall for drinkable water would be critical for the real-world performance of the system, ensuring that it can safely and accurately identify water that is suitable for consumption.

The obtained results with the modified SMOTE code were as follows:

Best Parameters from GridSearch: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}

Confusion Matrix:

274	120
130	276

Table A7 represents the classification Report (see the Appendix).

Concerning model improvement when introducing SMOTE (Synthetic Minority Over-sampling Technique) and hyperparameter tuning, here's an overview of how these techniques work and how they might have improved the model accuracy.

• Application of SMOTE

Typically, SMOTE is used to oversample the minority class in a binary classification task. For instance, in the context of predicting drinking water potability, if the dataset has more "Not Drinkable" samples than "Drinkable" samples, SMOTE would create synthetic examples of the "Drinkable" class.

This ensures that both classes (e.g., Potability = 0 for "Not Drinkable" and Potability = 1 for "Drinkable") are equally represented, helping the model learn better to distinguish between both classes.

• Hyperparameter Tuning

Hyperparameter tuning involves adjusting the settings (hyperparameters) of the ML model to optimize its performance. In this case, for RandomForestClassifier, hyperparameters such as number of estimators (trees), learning rate, and other model-specific parameters can be fine-tuned.

By using SMOTE and hyperparameter tuning together, the model's performance has been improved in several ways:

This combination can lead to:

Higher accuracy on the test set, as the model is better equipped to learn from the minority class (thanks to SMOTE) and make more informed predictions (thanks to hyperparameter optimization). Improved precision and recall for both classes, particularly for the minority class (e.g., "Drinkable" water samples), where the model may have struggled without SMOTE.

The obtained results with the modified SMOTE code and hyperparameter tuning are as follows:

We obtained the following results:

Best Parameters: 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200} Accuracy: 0.72625.

Confusion Matrix:

287	107
112	294

Table A8 represents the classification Report (see the Appendix).

By using SMOTE for class balancing and applying GridSearchCV to tune the hyperparameters, the model was able to improve overall accuracy to 72.6%, achieve a balanced performance across both classes (not drinkable and drinkable), with precision, recall, and F1-scores close to 0.73 and avoid overfitting by selecting appropriate hyperparameters, leading to better generalization.

These results reflect a significant improvement in performance due to the combination of class balancing (via SMOTE) and hyperparameter optimization (via GridSearchCV).

Feature engineering, scaling, and normalization were performed in the study to improve the effectiveness of both the Random Forest classifier and the TOPSIS decision-making process. The key steps taken are as follows:

Feature Engineering: The dataset included multiple physicochemical properties of water (e.g., pH, turbidity, hardness). Irrelevant or redundant features were removed to focus on critical parameters affecting potability. The TOPSIS ranking score was added as an additional feature, integrating decision-making into the ML process.

Data Scaling & Normalization (Random Forest): Random Forest does not require explicit feature scaling because it is tree-based, and splits data based on thresholds rather than distances.

For TOPSIS: Feature normalization (min-max scaling) was applied to ensure all variables had a common scale. Each feature was transformed as:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

This prevented features with larger numerical ranges from dominating the decision-making process.

(3) Handling Missing Values

Missing values were replaced with the mean of the respective feature to maintain consistency.

The confusion matrix analysis revealed that most misclassifications occurred in samples near the decision boundary particularly those with physicochemical values close to the threshold limits defined by WHO standards.

False positives (undrinkable samples predicted as drinkable) were primarily associated with samples that had borderline pH or nitrate levels, which, although not severely non-compliant, leaned toward unsafe levels. These cases highlight the challenge of modeling slight deviations in multi-parameter data where no single feature dominates the classification.

False negatives (drinkable samples predicted as undrinkable) typically occurred in cases with safe values overall but one or two atypical parameters—such as slightly elevated turbidity—which could have influenced the model's decision disproportionately. This suggests that further feature scaling or weighting could be investigated in future work.

4. Design of a robotic system

A robotic system has been designed and developed for the application of simulation software, previously [20,21]. The electronic devices we used were DC thruster or drive motors, battery, solar panel and DC/DC converter. Figure 4 shows a) the drive motors b) battery c) solar panel and d) DC/DC converter and Figure 5 shows remote control of the system prototype.

As shown in Figure 4, the electronic components required in the robotic systems including drive motors, battery, solar panel and DC/DC converter were used in the developed system.

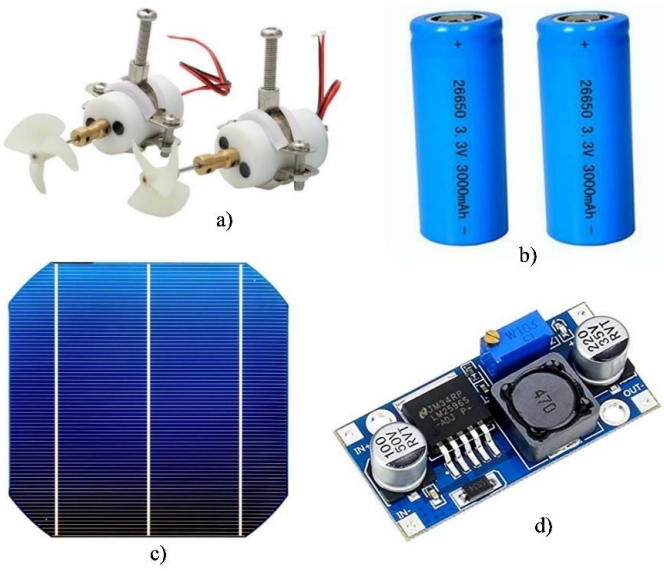

Figure 4. The electronic components of the robotic system for water analysis.

Another part related to the motors is their control. Control is done via a remote control and a command receiver (Figure 5).

As shown in Figure 5, 100 meters long distance for controlling the speed and direction of the motor, the remote control has four channels, and an adjustable speed, the direction can be adjusted well and easily to make the model go straight, the information receiving board at the input end has reverse connection protection, the output has a self-recovery fuse, frequency: 2.4G high frequency, to adapt and stable signal.

The connection between the battery and the motors is made to the receiver or ESC as is depicted in Figure 6.

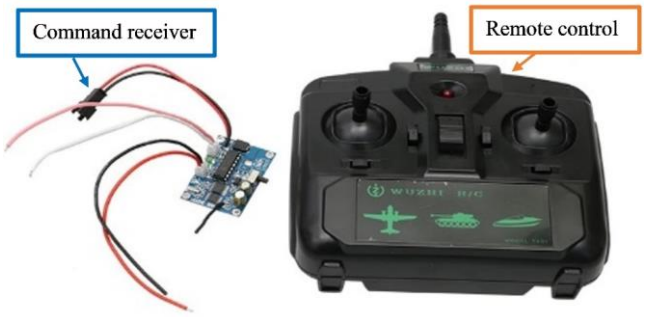

Figure 5. The remote control of the robotic system prototype for water analysis.

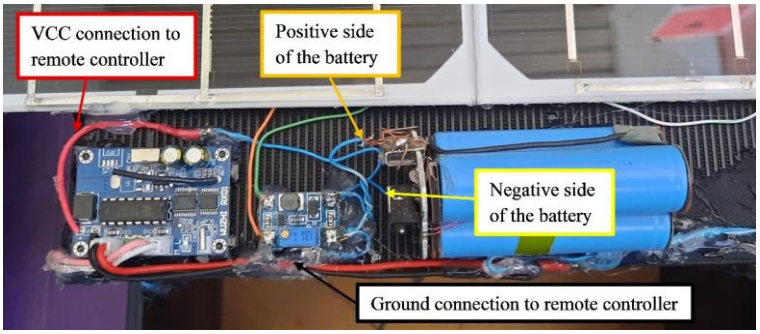

Figure 6. Remote controller receiver on the board.

As shown in Figure 6, the connection for the battery supply and the connection of the motors are made. Also, through the receiver, we can control the direction and speed of the motors. The receiver is a circuit that manages to transfer the energy obtained from the batteries to the motors and make them function. It has various capacitors for energy accumulation and other electrical elements for protection against current and voltage overloads, *etc.*

Figure 7 shows a prototype of a ship that could serve navigation and collection of water samples.

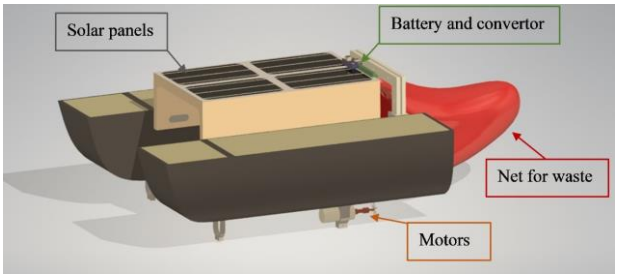

Figure 7. Ship prototype.

The designed system would be able to deposit the water samples in the storage area where they are collected by steering the ship via a robotic arm. The equipment needed to build the ship's platform contains motors, battery, solar panel, DC/DC converter, robotic arm, and sensors for monitoring the physical and chemical parameters of water (Figure 7).

In addition, we describe the components working together with the ML model to perform water analysis. The robotic system collects water samples using the motorized platform and robotic arm. Sensors measure the physicochemical parameters of the samples in real time. These sensor readings are processed by the onboard system, where the trained ML model classifies the water samples as drinkable or undrinkable. The classification results guide the robot's decision-making process, such as storing the sample or discarding it, thereby integrating hardware control with intelligent analysis.

In this study, hyperparameter tuning was not performed before using SMOTE (Synthetic Minority Over-sampling Technique) or Grid Search. However, we can address the general considerations and explain the approach taken:

• Hyperparameter Tuning

The Random Forest model used in the study was trained with default hyperparameters. These parameters, such as the number of estimators (n_estimators = 100), were chosen based on common practice and initial experimentation to ensure the model performed well on the dataset.

Hyperparameter tuning could be an important step to improve the model's performance, especially regarding max_depth, min_samples_split, and min_samples_leaf, which can affect complexity and overfitting.

• SMOTE

SMOTE was not used in the initial phases of the study. The dataset was assumed to be well-balanced based on the existing distribution of "Potability" (drinking *vs.* undrinkable water samples).

If SMOTE were to be used, it would be applied after hyperparameter tuning and cross-validation, ensuring that the synthetic data generated is used to train a well-tuned model.

In the current work, class imbalance was not explicitly addressed from the start. The dataset was assumed to have a balanced distribution of the "Potability" (drinking *vs.* undrinkable water) classes, and no specific steps were taken in the early stages to address class imbalance. Future work could explore the impact of class imbalance by incorporating techniques like SMOTE, class weighting, or other balancing methods to improve model accuracy and robustness.

Figure 8 shows the flowchart including data input, TOPSIS, ML and output applied in the current work for water analysis for a further application in robotics.

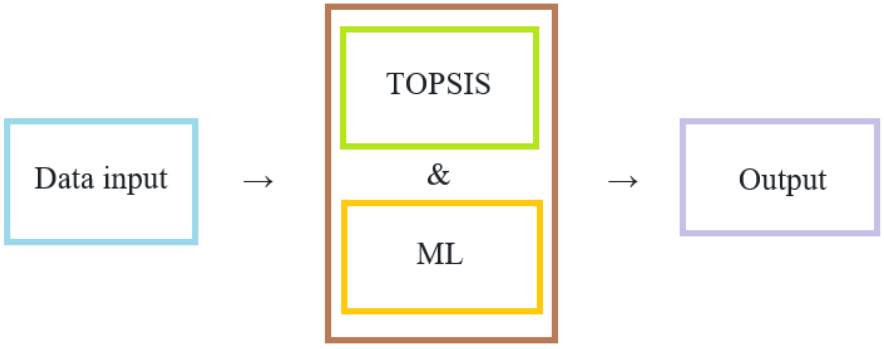

Figure 8. The flowchart including data input, TOPSIS, ML and output.

As shown in Figure 8, data input is followed by the application of combined TOPSIS and ML, which leads to the obtainment of output. In the current work, these approaches have been used together to improve the analysis results.

Table 5 shows the comparative results of the methodology used in the current work with the previous ones.

The current investigation following the previous published works has the advantages below in comparison with the other works. 1) The previous investigations for the current work have been based on modeling. The implementation of data using the developed model (model of the tree) have led to a better understanding of the obtained results with TOPSIS [10,11,17]. 2) The combination of TOPSIS and ML in the current work has led to the improvement of the water sample analysis. 3) The design and development of a new robotic system has helped a better investigation for water analysis with TOPSIS and ML and application in further works.

Table 5. The comparative methods and results of the current investigation and previous works.

Authors	Methods	Results
Hu XC, *et al.* [32]	ML	Prediction of water contaminants
Rajitha A, *et al.* [33]	ML	Simulation for water analysis
Gupta S, *et al.* [34]	Design of a robotic system	Design of a robotic system without water analysis
Javanbakht T [10,17]	Unmodified and modified TOPSIS	Decision making for analysis of water samples based on modeling
Javanbakht T [11]	Unmodified and modified TOPSIS	Automated decision making for analysis of water samples
Javanbakht T, Pajaziti A, Buza S (current work)	TOPSIS combined with ML, design and development of a robotic system	Improved water analysis with design and development of a robotic system

As shown in Table 5, several topics such as the prediction of water contaminants with ML, simulation for water analysis with this approach, design of a robotic system without water analysis have been performed, previously. Moreover, two recent studies showed the efficacy of TOPSIS for the prediction and ranking of water samples [10,11,17].

Further investigations including the application of sensors would be required for the implementation of the results of the current work in the developed robotic system. The next step of this study would be to use the ion detection sensors for detecting water contaminants and distinction of undrinkable and drinking water by a robotic system. Chemical sensors will be used to measure pH, turbidity, dissolved oxygen, and salinity. Conductivity meters will help determine water salinity and ion concentration. Total dissolved solids sensors will be applied to measure the concentration of dissolved particles. Also, microbial sensors will detect harmful bacteria and pathogens like E. coli and coliforms, which may be available in water samples [35].

In real-world applications, several factors can impact on the accuracy and reliability of water analysis using a robotic system. The main concerns include sensor accuracy, data noise, and scalability.

• Sensor accuracy

The precision of water quality sensors can vary due to environmental factors such as temperature fluctuations, sensor aging, and calibration issues. Some physicochemical parameters, such as pH and turbidity, require frequent recalibration to maintain accuracy. To mitigate errors, sensor fusion techniques or redundancy (using multiple sensors for the same parameter) can be implemented.

• Data noise

Real-world water samples can contain unpredictable variations caused by contaminants, sensor drift, or measurement inconsistencies. Preprocessing techniques such as data smoothing, filtering (e.g., Kalman or moving average filters), and anomaly detection can help reduce noise. The TOPSIS method and ML algorithms can be optimized to handle noisy data by incorporating feature selection and robust training datasets.

• Scalability for Large-Scale Water Analysis

As the dataset size increases, computational efficiency becomes a challenge, particularly when processing high-dimensional physicochemical data. The scalability of Random Forest and TOPSIS depends on the number of samples and features. While Random Forest can handle large datasets relatively well, TOPSIS may require optimization techniques such as parallel computing or dimensionality reduction (e.g., PCA). In a large-scale deployment, a distributed computing approach (such as cloud-based data processing) can help manage extensive water quality data collected from multiple robotic units. While the robotic system shows potential for real-world application, its deployment in industrial or rural environments may face challenges such as varying water conditions, limited connectivity, energy constraints, and the need for ruggedization of components. To address these challenges, future work will focus on improving sensor calibration, implementing noise-reduction techniques, and optimizing computational efficiency for large-scale water analysis. The integration of AI-based adaptive learning models could also enhance robustness and accuracy.

To address the growing need for efficient water resource management and exploration, it would be required to develop advanced robotic systems equipped with autonomous water sample analysis capabilities. These enhancements will be realized through the integration of cutting-edge technologies in robotics, sensor systems, and AI. Robots will be designed for diverse terrains, such as urban environments, rural areas, or remote planetary landscapes. Compact, mobile units will enable accessibility to challenge locations, including disaster zones or extraterrestrial surfaces. Autonomous sampling and self-optimization will be developed for the robotic system. Robots will autonomously identify and extract water samples from natural or artificial reservoirs. Utilizing AI, they will dynamically adjust their processes to maximize efficiency in sampling and analysis. The system will include wireless communication modules to relay findings to human operators or centralized systems. A user-friendly interface will provide real-time feedback and actionable insights for decision-makers.

The benefits of autonomous water sample analysis robots will be as below: 1) Rapid Response in Crisis Situations: During natural disasters or environmental crises, robots can quickly identify safe drinking water sources, aiding emergency response efforts. 2) Sustainable Resource Management: Robots can monitor water bodies regularly, ensuring early detection of pollution or resource depletion. 3) Planetary Exploration: Autonomous robots can play a vital role in space missions by identifying potential water sources on extraterrestrial surfaces, contributing to the search for life and the establishment of habitable zones. 4) Reduction in Human Intervention: These robots minimize the need for human presence in hazardous or inaccessible areas, reducing risks and increasing efficiency.

The challenges related to sensor accuracy, data noise, and system scalability provide a balanced perspective on the feasibility of our robotic water analysis system by highlighting both potential limitations and strategies for overcoming them. Addressing these issues ensures that the system remains practical and effective for real-world applications. Despite the challenges, our results demonstrate that

combining decision-making (TOPSIS) with ML (Random Forest) enhances the system's ability to analyse water quality effectively. With continuous improvements in sensor technology, data processing, and robotic control, the proposed system is feasible for both small-scale and large-scale applications.

Random Forest was selected for ML in this study due to its robustness, accuracy, and ability to handle complex datasets in water analysis. Below are the key reasons for this choice:

Handling of Nonlinear Relationships and Complex Data: Water quality datasets often include nonlinear relationships between physicochemical properties (e.g., pH, turbidity, and contaminant concentration). Random Forest, being an ensemble learning method, builds multiple decision trees and combines their results, making it effective in capturing complex patterns in water quality data.

Resilience to Noisy and Incomplete Data: Real-world water quality data can contain missing values or noise due to sensor inaccuracies. Random Forest can handle missing data by using surrogate splits and is less sensitive to noise compared to single decision trees.

Feature Importance for Decision-Making: The algorithm provides insights into the importance of each water quality parameter in classification. This helps in identifying the most critical factors for distinguishing drinking water from non-drinkable samples.

Generalization and Overfitting Prevention: Unlike traditional decision trees, Random Forest reduces overfitting by averaging predictions from multiple trees. This ensures that the model generalizes well when applied to new water quality samples.

Performance and Interpretability: The high classification accuracy (69% in our first step of simulations) demonstrated the algorithm's effectiveness. Additionally, Random Forest provides interpretable results, making it easier to analyze and validate predictions.

The results presented in this work can lead to the application of robots for analysis and distinction of drinking and undrinkable water on Earth and other planets based on autonomous water testing in remote areas and early contamination alerts.

5. Conclusion

This work focused on the improvement of robot learning for the optimization of water analysis with the decision-making and ML processes for a robotic system. The combined algorithms were used for several series of analysis. The results with TOPSIS showed the impact of the high values of profit criteria and low values of cost criteria on the candidates' ranks. This was obtained in two different analyses of the water physicochemical properties and ingredient concentrations. The differences in the closeness coefficient values of the best and the worst candidates in the first and second analyses were 35%, and 45%, respectively. These analyses were made on different numbers of candidates considering different criteria in each series of analysis. In the first series of analysis, the best ranked and the worst ranked water samples were differentiated with a 35% difference in their closeness coefficient values. In the second series of analysis, the difference of these values for the best and the worst ranked water samples reached 45%, which revealed the appropriate candidates' ranking with TOPSIS. The ML simulation showed the improvement of the learning accuracy to 69% using the modified code. After using SMOTE for class balancing and applying GridSearchCV to tune the hyperparameters, the model was able to improve overall accuracy to 73%, achieving a balanced performance across both classes (not drinkable and drinkable). Further investigations would be carried out using a robotic system for which the electronic components and the remote control of its prototype have been designed and developed. The

obtained results could be used for further development of robot learning in science and engineering for water analysis.

Authors' contribution

Conceptualization of water analysis, T.J.; methodology, T.J. and A.P.; writing—original draft preparation, T.J.; revised manuscript preparation, T.J. and A.P.; investigation, T.J. and A.P.; data curation, T.J.; robotic system design: S.B.; robotic system development, A.P. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

The authors declare no conflict of interest.

Abbreviations

AI: Artificial intelligence; TOPSIS: Technique for Order of Preference by Similarity to Ideal Solution; ML: Machine learning; SMOTE: Synthetic Minority Over-sampling Technique.

Appendix

(A) Tables

Table A1. Mean values of fuzzy data for water samples.

Candidates/ Criteria	Calcium	Magnesium	Boron	Copper	Cadmium	Chromium	Arsenic	Nickel	Lead	Nitrite	Sodium	Chloride	1,2-Dichlorobenzene	Monochlo-ramine	Dichloroiso-cyanurate	Toluene
C_1	0.8	0.6	0.02	0.02	0.00003	0.0005	0.0001	0.0007	0.0001	0.5	0.05	0.03	0.01	0.02	0.4	0.01
C_2	0.2	0.2	0.02	0.02	0.00003	0.0005	0.0001	0.0007	0.0001	0.5	0.05	0.03	0.01	0.02	0.4	0.02
C_3	0.2	0.2	0.04	0.04	0.00005	0.001	0.0004	0.0028	0.0004	0.1	0.1	0.06	0.03	0.04	0.1	0.02
C_4	0.4	0.3	0.04	0.08	0.00001	0.002	0.0002	0.0014	0.0002	0.1	0.1	0.01	0.01	0.02	0.2	0.04

Table A2. Weight values applied for each water sample.

Alternatives/ Values	Calcium	Magnesium	Boron	Copper	Cadmium	Chromium	Arsenic	Nickel	Lead	Nitrite	Sodium	Chloride	1,2-Dichlorobenzene	Monochlo-ramine	Dichloroiso-cyanurate	Toluene
C_1–C_4	0.1	0.1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.1	0.1	0.05	0.05	0.05	0.05

Table A3. The criteria matrix for water samples.

Alternatives/ Values	Calcium	Magnesium	Boron	Copper	Cadmium	Chromium	Arsenic	Nickel	Lead	Nitrite	Sodium	Chloride	1,2-Dichlorobenzene	Monochlo-ramine	Dichloroiso-cyanurate	Toluene
C_1–C_4	True	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Table A4. The distances from the ideal solutions, the similarity coefficients and the ranking of candidates.

Candidates	di*	di⁻	CC_i	Ranking
C_1	0.04828959	0.13201336	0.73217526	1
C_2	0.09768143	0.09908455	0.50356547	4
C_3	0.13564753	0.05183078	0.2764628	2
C_4	0.0901123	0.09844483	0.5220955	3

Table A5. Classification report.

	Precision	Recall	f1-score	Support
0	0.70	0.88	0.78	412
1	0.63	0.35	0.45	244
Accuracy	0.68			656
Macro avg	0.66	0.61	0.61	656
Weighted avg	0.67	0.68	0.66	656

Table A6. The characteristic data.

pH	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic carbon	Trihalomethanes	Turbidity	Potability	TOPSIS score	Model prediction
9.014357	225.367519	49009.924660	6.002653	296.630899	445.768812	16.254112	83.891129	4.549419	1	0.645133	Drinkable
9.097617	263.824120	38413.057090	7.634362	274.959028	437.263531	18.285882	74.512565	5.473663	1	0.644199	Drinkable
9.079715	222.042631	53735.899190	6.894915	254.040977	382.896166	12.704887	101.615346	4.551724	1	0.631491	Drinkable
9.961503	276.699765	18743.222490	9.160740	361.221165	457.551559	21.564489	93.740334	5.215590	0	0.617467	Undrinkable
6.563357	241.286323	56320.586980	5.365558	333.775777	415.817219	11.651929	70.637648	5.292950	1	0.615420	Drinkable

Table A7. The classification report.

	Precision	Recall	f1-score	Support
0	0.68	0.70	0.69	394
1	0.70	0.68	0.69	406
Accuracy	0.69		0.69	
Macro avg	0.69	0.69	0.69	800
Weighted avg	0.69	0.69	0.69	800

Table A8. The classification report.

	Precision	Recall	f1-score	Support
0	0.72	0.73	0.72	394
1	0.73	0.72	0.73	406
Accuracy	0.73		800	
Macro avg	0.73	0.73	0.73	800
Weighted avg	0.73	0.73	0.73	800

(B) The combined TOPSIS and ML quasi-code below was used to classify water samples as drinkable or undrinkable.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from topsis import Topsis # Ensure you have a TOPSIS implementation
# Set display options to show all values
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.float_format', '{:.6f}'.format)
np.set_printoptions(threshold = np.inf)
# Load dataset
```

```
file_path = "water_quality.csv"
df = pd.read_csv(file_path)

# Handle missing values (replace with mean of column)
df.fillna(df.mean(), inplace=True)

# Extract features and target
X = df.drop(columns = ['Potability'])
y = df['Potability']

class Topsis:
 def __init__(self, data, weights = None, criteria = None):
 self.data = data
 self.weights = weights
 self.criteria = criteria
 # Your logic here
import numpy as np


# Normalize the feature matrix for TOPSIS
X_normalized = (X - X.min()) / (X.max() - X.min())

# Define weights and criteria (+ for benefit, - for cost)
weights = np.ones(X.shape[1]) / X.shape[1] # Equal weights
criteria = ['+'] * X.shape[1] # All features considered beneficial

# Apply TOPSIS ranking
topsis_model = Topsis(X_normalized, weights, criteria)
topsis_model.calc()
rankings = topsis_model.rank_to_best_worst()

# Add rankings to dataset
df['Topsis_Score'] = rankings



# Train-test split
X_train, X_test, y_train, y_test = train_test_split(df.drop(columns=['Potability']), y, test_size=0.2,
random_state=42)

# Train Machine Learning model
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
clf.fit(X_train, y_train)

# Predictions
y_pred = clf.predict(X_test)

# Print results
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)

# Get decision matrix (features only, no target)
decision_matrix = df.drop(columns=['Potability']).values

# Now, integrate the model's predictions and the TOPSIS ranking
df_sorted['Model_Prediction'] = np.where(df_sorted['Potability'] == 1, 'Drinkable', 'Not Drinkable')

# Show the final output with model predictions and TOPSIS ranking
print(df_sorted[['Potability', 'Topsis_Score', 'Model_Prediction']].head())
```

The improved code was as below.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from imblearn.over_sampling import SMOTE
# Load dataset
file_path = "water_quality.csv"
df = pd.read_csv(file_path)

# Handle missing values (replace with mean of column)
df.fillna(df.mean(), inplace=True)

# Extract features and target
X = df.drop(columns=['Potability'])
y = df['Potability']

# Apply SMOTE for class balancing
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
# Train-test split
```

```
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2,
random_state=42)

# Define parameter grid for hyperparameter tuning
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]  # Adjusted terminology to reflect water samples
}

# Initialize classifier
clf = RandomForestClassifier(random_state=42)

# Perform GridSearchCV
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Get best model
best_clf = grid_search.best_estimator_

# Predictions
y_pred = best_clf.predict(X_test)

# Evaluate model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Print results
print("Best Parameters:", grid_search.best_params_)
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

References

[1] Chen LY, Xu C, Dharmarajan K, Irshad MZ, Cheng R, *et al*. RoVi-Aug: robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv* 2024, arXiv:2409.03403.

[2] Seraj E, Lee KM, Zaidi Z, Xiao Q, Li Z, *et al.* Iterative and explainable robot learning: a comprehensive review. *Found. Trends Robot.* 2024, 12(2–3):73–349.

[3] Kyrarini M, Haseeb MA, Ristić-Durrant D, Gräser A. Robot learning of objective manipulation task actions from human demonstrations. *Facta Univ., Ser. Mech. Eng.* 15(2):217–229.

[4] Chernova S, Thomaz AL. *Robot learning for human teachers*, 1st ed. San Rafael: Morgan & Claypool Publishers, 2014.
[5] Chetouani M. Interactive robot learning: an overview. In *Human-Centered Artificial Intelligence*, 1st ed. Cham: Springer, 2023. pp. 140–172.
[6] Gidlow DA. Lead toxicity. *Occupational Med.* 2015, 65(5):348–356.
[7] Ranjbar L, Yamini Y, Saleh A, Seidi S, Faraji M. Ionic liquid based dispersive liquid-liquid microextraction combined with ICP-OES for the determination of trace quantities of cobalt, copper, manganese, nickel and zinc in environmental water samples. *Microchimica Acta* 2012, 177 (1–2):119–127.
[8] Tian M, Fang L, Yan X, Xiao W, Row KH. Determination of heavy metal ions and organic pollutants in water samples using ionic liquids and ionic liquid-modified sorbents. *J. Anal. Methods Chem.* 2019, 2019(1):1948965.
[9] Tang DWS. Phreatic zone wastewater irrigation: sensitivity analysis of contaminant fate. *J. Hydrol.* 2024, 645B:132263.
[10] Javanbakht T. Automated decision-making with TOPSIS for water analysis. *J. Eng. Sci.* 2022, 9(1):H19–H24.
[11] Javanbakht T. A novel automated decision-making process for analysis of ions and organic materials in drinking water. *J. Eng. Sci.* 2023, 10(1):H1–H7.
[12] Zulqarnian RM, Xin XL, Saqlain M, Khan WA. TOPSIS method based on the correlation coefficient of interval-valued intuitionistic fuzzy soft sets and aggregation operators with their application in decision-making. *J. Math.* 2021, 10:1–16.
[13] Yuvaraj N, Kumar MP. Multiresponse optimization of abrasive water jet putting process parameters using TOPSIS approach. *Mater. Manuf. Processes* 2015, 30:882–889.
[14] Shyur HJ, Yin L, Shih HS, Cheng CB. A multiple criteria decision making method based on relative value distances. *Found. Comput. Decis. Sci.* 2015, 40(4):299–315.
[15] Sodhi B, Tadinada P. A simplified description of Fuzzy TOPSIS. *arXiv* 2015, arXiv:1205.5098.
[16] Dutta B, Singba T, Goh M, Lamata MT, Verdegay JL. Post factum analysis in TOPSIS based decision making method. *Expert Syst. Appl.* 2019, 138:112806.
[17] Javanbakht T. Modélisation et traitement informatique de l'inconsistance des croyances épistémiques. Doctoral Thesis, University of Quebec in Montreal, 2022.
[18] Fouad H, Moskowitz IS, Brock D, Scott M. Integrating expert human decision-making in artificial intelligence applications. In *Human-Machine Shared Contexts*, 1st ed. London and Boston: Academic Press, 2020. pp. 257–275.
[19] Louis M, ElAzab M. The future of teaching and learning in artificial intelligence era (part II). *Int. J. Internet Educ.* 2023, 22(2):1–8.
[20] Pajaziti A, Basholli F, Zhavelli Y. Identification and classification of fruits through robotic system by using artificial intelligence. *Eng. Appl.* 2023, 2(2):154–163.
[21] Likaj R, Bajrami X, Shala A, Pajaziti A. Path finding for a mobile robot using fuzzy and genetic algorithms. *Int. J. Mech. Eng. Technol.* 2017, 8(8):659–669.
[22] Pajaziti A, Stuja K, Gjelaj A, Calvin L. Design and optimization of the welding robot system for aluminium frames using simulation software. In *Proceedings of the 34th International DAAAM Symposium 2023*, Online, October 26–27, 2023, pp. 40–49.

[23] Li K. The influence of Python on logical thinking and analysis of teaching countermeasures. *J. Phys. Conf. Ser.* 2021, 1744(4):042164.

[24] Cutting V, Stephen N. A review on using Python as a preferred programming language for beginners. *Int. Res. J. Eng. Technol.* 2021, 8(8):4258–4263.

[25] Ullah I, Adhikari D, Khan H, Anwar S, Ahmad S, *et al*. Mobile robot localization: current challenges and future perspectives. *Comput. Sci. Rev.* 2024, 53:100651.

[26] Ullah I, Adhikari D, Khan H, Ahmad S, Esposito C, *et al*. Optimizing mobile robot localization: drones-enhanced sensor fusion with innovative wireless communication. In *IEEE INFOCOM 2024—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Vancouver, Canada, May 20, 2010, pp. 1–6.

[27] Khan S, Ullah I, Ali F, Shafiq M, Ghadi YY, *et al*. Deep learning-based marine big data fusion for ocean environment monitoring: towards shape optimization and salient objects detection. *Front. Mar. Sci.* 2023, 9:1094915.

[28] Ullah H, Ali F, Sharafian A, Ali A, Naeem HMY, *et al*. Optimizing underwater connectivity through multi-attribute decision-making for underwater IoT deployments using remote sensing technologies. *Front. Mar. Sci.* 2024, 11:1468481.

[29] Hussain A, Hussain T, Ullah I, Muminov B, Khan MZ, *et al*. CR-NBEER: cooperative-relay neighboring-based energy efficient routing protocol for marine underwater sensor networks. *J. Mar. Sci. Eng.* 2023, 11:1474.

[30] Hassan R, Abubakar ZM, Usman NS, Barde SY, Salau HB, *et al*. Dynamic neighbor adaptation-based energy-efficient routing protocol for underwater wireless sensor networks. In *2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON)*, Ado Ekiti, Nigeria, November 26–28, 2024, pp. 1–5.

[31] Tharmalingam L. Water quality and potability [dataset]. 2024. https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability/data (accessed on 5 February 2025).

[32] Hu XC, Dai M, Sun JM, Sunderland EM. The utility of machine learning models for predicting chemical contaminants in drinking water: promise, challenges, and opportunities. *Curr. Environ. Health Rep.* 2023, 10:45–60.

[33] Rajitha A, Aravinda K, Nagpal A, Kalra R, Maan P, *et al*. Machine learning and AI-driven water quality monitoring and treatment. In *3rd International Conference on Applied Research and Engineering (ICARAE2023)*, Cape Town, South Africa, November 17–19, 2024, p. 03012.

[34] Gupta S, Kohli M, Kumar R, Bandral S. IoT based underwater robot for water quality monitoring. In *International Conference on Integrated Interdisciplinary Innovations in Engineering (ICIIIE 2020)*, Chandigarh, India, August 28–30, 2020, p. 012013.

[35] Maheux AF, Huppé V, Boissinot M, Picard FJ, Bissonnette L, *et al.* Analytical limits of four β-glucuronidase and β-galactosidase-based commercial culture methods used to detect Escherichia coli and total coliforms. *Methods* 2008, 75(3):506–514.