International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–14

http://dx.doi.org/10.12785/ijcds/1571031447

Hybrid Intelligent Technique between Supervised and Unsupervised Machine Learning to Predict Water Quality

Hanan Anas Aldabagh[1] **and Ruba Talal Ibrahim**[2]

[1]*Department of Computer Science,The General Directorate of Education in Nineveh Governorate,Mosul,Iraq*
[2]*Department of Computer Science,University of Mosul, Mosul,Iraq*

Received 17 May 2024, Revised 5 December 2024, Accepted 5 December 2024

Abstract: Water is the secret of life and occupies over 70% of the Earth's surface. It has become necessary to protect the water resources around us from pollution and neglect, which can result in the loss of life and health. Artificial Intelligence (AI) has the potential to improve water quality analysis, forecasting, and monitoring systems for sustainable and environmentally friendly water resource management. As a result, this work focuses on the multi-model learning features to represent the state of the water and determine its suitability category (i.e., safe or unsafe). This is done by building a jointly hybrid model between supervised algorithms and unsupervised algorithms after fusing their outliers. In addition, the Gamel herd swarm optimization algorithm was applied to find the optimum hyperparameters.Two datasets were used, in the first dataset the proposed hybrid model outperformed other models by 99.2% in accuracy, AUC, and f1 score, but in the second dataset, it achieved approximately 92% in accuracy, AUC, and f1-score. Finally, the paper offered a methodology that researchers can use to anticipate water quality using hybrid machine learning.

Keywords: Prediction, Water Quality, Artificial Intelligent, Machine Learning, Supervised, Unsupervised, Camel Herd Algorithm.

1. INTRODUCTION

Water is an important vital resource for sustaining the life of living organisms, since clean water is used in various aspects of life, such as drinking, agriculture, energy generation, and entertainment [1]. But because of modern technology, aquatic ecosystems have polluted the water and the species that live in it. Most countries in the world use chemicals in addition to selling them, which exposes water to toxic substances and makes it unfit for consumption by living organisms as well as agriculture. The sustainability of all living things in the context of the new green economy depends critically on the monitoring and analysis of water quality [2]. Due to the existence of precise water quality standards, traditional chemical monitoring methods cannot assess the complex interactions and impacts of many chemicals on microorganisms in water [3]. Many organizations employ manual techniques to monitor water quality and assess complicated interactions, calculating the water quality index (WQI) equation after collecting samples and analyzing them in a laboratory, which has proven to be costly and time-consuming. Recently, many AI studies have demonstrated the possibilities of employing ML technology and sensors to process the problem of forecasting water quality ,consumption and automating their monitoring, as well as the ability to collect data in real time [4],[5]. ML, a subfield of AI, allows a system to automatically learn and train data in order to recognize trends and update itself without the need for explicit programming [6]. ML opens up new prospects for predicting WQI in water body investigations by giving photo-sensors that based on calculating the wavelength of a given color or variations in amplitude values, which may be utilized to detect various dissolved water contaminants [7],[8].The outputs of these sensors can generate data that is processed using ML techniques with high accuracy and performance. ML models may successfully mimic hydrological processes and pollution transport when big datasets are available [9].

In this paper, a ML was used to predict the quality of water whether it is suitable for drinking or not, instead of traditional expensive methods that require time and many efforts. A jointly hybrid technique was applied that combines supervised ML methods and unsupervised ML methods on a survey online dataset (Kaggle) and outperformed previous studies. The following contributions were made:

1) The dataset was processed by using normalization and oversampling to balance it.

2) The Light Gradient Boosting Model (LGBM)

E-mail address: hanan_aldabagh1985@nan.epedu.gov.iq, rubatalal@uomosul.edu.iq

hyper-parameter's were tuned using the Gamel herd method.

3) The main contribution of water quality prediction is whether it is suitable for use or not by using the original features in the dataset, as well as new features (outliers) extracted from unsupervised ML methods Copula-Based Outlier Detection (COPOD),Isolation Forest (IForest),and Cluster-based Local Outlier Factor(CBLOF) and forming combined features that are passed to LGBM technique to perform the final prediction process.

4) The performance of the proposed models was evaluated using a number of performance metrics (accuracy, precision, recall, F1 score, AUC-ROC).

5) A comparison was done between LGBM technique after balancing the dataset and the hybridized LGBM with unsupervised (COPOD, IForest, and CBLOF) ML approaches

6) A comparison was done between LGBM technique with traditional ML.

7) Finally, the proposed model was compared to the previous studies.

The rest of work organized as follow: section 2 will discuss related work, section 3 and 4 will present structure of supervised and unsupervised ML, section 5 will discuss Gamel herd algorithm, section 6 will present description and analysis of the dataset, section 7 will present correlation analysis. Finally, section 8 will present Research Methodology and results discussion followed by section 9, which is conclusion.

2. Related Work

Several previous studies have validated the use of AI algorithms for water quality prediction and analysis. Here's a summary of these studies:

Furqan Rustam et al.[10] reviewed ML techniques to improve the prediction of water consumption and quality, using two types of unbalanced datasets, the first from the Gaggle website to predict water quality and the second from GitHub to predict water consumption. The limitation of this study was unbalanced datasets. After tuning the hyperparameters, paper employed a variety of ML methods. This study improved Artificial Neural Network (ANN) Model after adding ReLU activation function followed by dropout layer with 50% dropout rate to reduce complexity and prevent overfitting. The ANN model was constructed up of three layers: the first and second layers each had 256 nodes, while the final layer had two nodes to predict water quality and one node to predict water consumption. The findings revealed that this study obtained an accuracy range of 90% to 99%, with an enhanced ANN outperforming the other models with an accuracy of 96% for forecasting water quality and a 99% R2 score for water usage. At the same time, Nida Nasir et al.[11] introduced study which involved a variety of twelve ML algorithms, including Suport Vector Machine(SVM), Random Forest(RF), Logistic Regression(LR), Decision Tree(DT), CATBoost, Extreme Gradient Boosting(XGBoost), and Multilayer perceptron(MLP), as well as an ensemble of all models. To estimate water quality, the paper analyzed data obtained from different Indian towns. The CatBoost method was considered the most dependable by the study, achieving 94.5% accuracy and producing 100% accuracy after ensemble the models.

Duie Tien Bui et al.[12]assessed the efficacy of four standalone algorithms :Random Forest (RF),Reduced Error Pruning Tree (REPT), Model Tree Algorithm (M5P), and Regression Tree (RT) and twelve hybrid data-mining algorithms(hybrids of standalones with CVPS, Bagging, and Random Forest Classifier (RFC)) in predicting WQI. The study relied on a dataset collected from northern Iran. The modeling procedure found that fecal coliform content was the most critical factor influencing WQI. The findings showed that the performance of the separate and hybrid models varied based on the differences in the input features (water samples). The features with the highest correlation coefficient had the most predictive power, and vice versa. The hybrid (BART) approach outperformed the other hybrid or standalone models, with an R2 score of 94%, although it may not perform as well in different datasets and environments.

Mohamed Torky et al.[13]presented ML techniques to predict whether drinking water samples are safe or dangerous, in addition to predicting WQI.Nine ML models were used to categorize water samples like RF and LGBM models which outperformed other models with accuracy rates of 96% and 97%, respectively. As for regression, six models were used to predict WQI, with superiority LGBM regression models and Extra tree regression models with an accuracy of 95.5% on the rest.

Fitore Muharemi et al.[14]employed time series data gathered by the General Water Company of Germany as a challenge to estimate water quality. The study used a variety of ML methods (LR,SVM, Linear Discriminant Analysis(LDA), recurrent neural network (RNN),ANN, Deep Neural Network (DNN), and Long Short-Term Memory (LSTM), and the findings revealed that imbalanced data has a significant impact on the performance of ML algorithms and makes them vulnerable. As a result, the paper did not produce satisfactory findings, particularly when applying time series algorithms (DNA, RNN, and LSTM). Meanwhile, Umair Ahmed et al.[15]classified water quality (WQC) and predicted WQI by applying a set of ML algorithms. The researchers collected dataset from several different sources for Lake Rawal in the city of Pakistan. The research relied on a number of important parameters after performing a number of preprocessing on them, such as temperature, pH, and others. The results demonstrated that gradient boosting and polynomial regression achieved best accuracy for predicting WQI while in water quality classification, the MLP model overcame the rest models

with an accuracy of 85%. To forecast water quality, Md. Mehedi Hassan et al. [16], applied several of supervised ML models in India. The research relied on a dataset collected from Kaggle consisting of a number of important biometric features that indicate water quality and purity. The findings showed that MLR outperformed the other models with about 99% accuracy. At the same year, M. H. Al-Adhaileh, and F. W. Alsaade [17] employed two approaches. The first approach was to use the created Adaptive Neural Fuzzy Inference System (ANFIS) algorithm to estimate WQI. The second is to use Feed-Forward Neural networks (FFNN) and K-Nearest Neighbors (KNN) to classify water quality. The analysis was based on seven major features, and after evaluation using a variety of performance indicators and statistics, the two models produced the best results.
Saber Kouadri [18] proposed two scenarios: in the first scenario, all parameters were utilized as inputs and tried to shorten the time required for WQI computation. In the second scenario, all inputs were decreased based on sensitivity analysis and aimed to illustrate the fluctuation in water quality in crucial instances where the required assessments are not available. The study employed eight AI algorithms to forecast water quality indicators in an arid desert setting using 114 samples taken at various time intervals from six aquifers in Illizi Province, southeastern Algeria. The findings revealed that the MLR model had the highest accuracy.
Afaq Juna et al. [19] predicted water quality based on data at the kaggle website after processing it, such as eliminating missing values using KNN imputer or manually. The work applied a number of traditional ML methods, in addition to improving the MLP model, which consists of nine layers, with 256 nodes in each layer. The model was implemented over 20 epochs and used the loss function(binary_crossentropy) with Adam Optimizer. The results showed that the improved model with KNN imputer achieved the best results with an accuracy of 99%. Table I summarized Related Work.

3. Supervised Machine Learning

Machine learning algorithms are trained using data that is labeled. Each data point includes input characteristics and their corresponding output labels.The LGBM algorithm is an example of supervised ML that was applied in this work. The algorithm was presented by Ke et al. [20] and based on Decision Tree algorithm. In comparison to traditional techniques, the algorithm's design, which combines Gradient-Based One-Sided Sampling (GOSS) and Exclusive Feature Pooling (EFB), offers high efficiency, accuracy, and regression in data classification [21]. GOSS relies on high gradients and leaves out features with low gradients. In order to minimize the amount of features, mutually incompatible features are bundled together using EFB [22]. It is characterized by:

1) It is called light because of its speed in training data.
2) Less memory consumption.
3) Reaching the best accuracy.
4) Dealing with big data.
5) Followed parallel learning
6) It reduces the cost of loss because it relies on dividing the tree into leaves and not at the depth level that used in previous Boosting algorithms.

4. Unsupervised Machine Learning

Unsupervised machine learning algorithms identify patterns and structures within data without requiring labels. Below the (COPOD, IForest, CBLOF) algorithms are an examples of unsupervised ML that was applied in this work.

A. Copula-Based Outlier Detection (COPOD)

It was introduced by Zheng Li [23], who characterized it as being motivated by copulas for modeling multivariate distributions. Copulas are mathematical functions that allow the COPOD model to distinguish marginal distributions from a random data. This offers COPOD the ability to be employed in high-dimensional datasets [24]. The method creates an empirical copula to estimate the tail probability of each data point and identify its "extreme" level.COPOD advantages are [23]:

- COPOD has no hyperparameters, is based on Empirical Cumulative Distribution Functions (ECDFs), and does not employ random learning or training. This eliminates the difficulty of selecting hyperparameters and potential biases.

- It can discover anomalies that impact the joint distribution of a set of variables, allowing it to work with multidimensional data and provide adaptable models of interactions among variables. It outperforms Principal Component Analysis (PCA), which is less successful at recognizing multidimensional anomalies since it depends on dimensionality reduction as well as data reconstruction.

- It may be more appropriate for data with many types of behaviors and distributions as it outperforms Density-based spatial clustering of applications with noise (DBSCAN) and K-Means, which might need more regular data and precise distributions to detect anomalies successfully. The working steps are [25]:

1) The dataset is collected, and then preprocessed to deal with missing and abnormal values.
2) The variables in the dataset are treated as having a uniform distribution using marginal distribution functions. Use the copula function like (Gaussian, Clayton etc.) to represent the dependence structure between the converted variables.
3) The parameters of the chosen copula function are determined using maximum probability estimation or other fitting approaches.
4) The copula function constructs synthetic data points that reflect the dependence structure of the original dataset.

TABLE I. Summarization of Related Work

Papers	Year	Methods	Dataset	Best Results
[10]	2022	DT, RF, Extra Tree, LR, AdaBoost, CNN, LSTM, Gated Recurrent unit and improved ANN	https://www.kaggle.com/datasets/adityakadiwal/water-potability	96% forecasting water quality 99% water Consumption
[11]	2022	SVM, LR, RF, DT, XGBoost CATBoost, and (MLP)	https://kaggle.com/anbarivan/indian-water-quality-data	Cat boost 95% 100% Meta decision tree, Meta MLP, Meta CATBoost
[12]	2020	M5P, RF, RT, REPT (reduced error pruning tree), BA (bagging)-M5P, BA-RT, BA-RF, CVPS (CV parameter selection)-M5P, RFC-RT, BA-REPT, CVPS-RT, CVPS-REPT, RFC-RF, RFC-M5P, RFC-REPT	Private	94% BA-RT
[13]	2023	XGBoost, Decision Tree, LGBM, MLP, ETC Classifier, ANN, GBC, RF, SVM	https://www.kaggle.com/datasets/mssmartypants/water-quality	Classification 96% RF 97% LGBM Regression 95.5% LGBM and DT
[14]	2019	LR, linear discriminant analysis, SVM, ANN, recurrent neural network (RNN), deep neural network (DNN), LSTM	Private	36% SVM
[15]	2019	Multiple Linear Regression, Ridge Regression, Polynomial Regression, Lasso Regression, Elastic Net Regression, RF, SVM, Gaussian Naïve Bayes, MLP, LR, Stochastic gradient descent, K Nearest Neighbor, DT, Bagging Classifier	http://www.pcrwr.gov.pk/	Classification (MLP) 85% accuracy Regression Gradient Boosting 7.2011 MSE polynomial regression 12.7307 MSE
[16]	2021	ANN, SVM, bagged tree (BT) models, RF, multinomial logistic regression (MLR)	Indian dataset pollution https://www.kaggle.com/code/anbarivan/indian-water-quality-data	MLR 100%
[17]	2021	Adaptive Neural Fuzzy Inference System (ANFIS), feed-forward neural networks (FFNN), K-nearest neighbors	Indian water quality data (kaggle.com)	ANFIS 92.39%accuracy FFNN 100% accuracy KNN 80.63% accuracy
[18]	2021	Multi linear regression (MLR), ANN, SVM, M5P tree, Random subspace(RSS), RF, Additive regression (AR), and Locally weighted linear regression (LWLR)	Private	MLR 100%
[19]	2022	LR,SVC, DT, RF,KNN,Stochastic Gradient Decent Classifier (SGDC), and XGBoost, MLP-9	https://www.kaggle.com/datasets/adityakadiwal/water-potability	MLP-9 99% accuracy

5) Data points that deviate significantly from the expected adoption structure are considered outliers.
6) Statistical analysis is utilized to describe the features of outliers and the causes for their anomaly.

B. Isolation Forest (IForest)

In 2008, Zhou Zhihua produced unsupervised IForest algorithm. It is an efficient and ensemble learning method that identifies outliers throughout the full sample space [26]. This method provides a good level of accuracy and execution efficiency. It may identify anomalous data by isolating data points that are sparse and dispersed from high density clusters. Following are some advantages of IForest [27]:

- It employs a large number of short-depth trees to extract outliers. This makes it very fast when dealing with large datasets, unlike other methods like k-means or DBSCAN, which may require extensive computations.
- It can recognize outliers without knowing how many them exist in the data, making it more adaptable than many other algorithms.
- Its utilization of quick and easy decision trees allows it to be readily extended to several applications and enormous datasets. The principle of its work is [28]:

1) A subset of the training data is chosen randomly.
2) iteratively creates binary trees, each branch of which is called an isolation tree (itree).
3) Each time, the feature and partition value(p) are selected at random, with the condition that the partition value (p) is within the feature value range. If feature ¡ p, then put it in the left tree otherwise put in the right tree.
4) The stopping condition for the algorithm is to reach the deepest node in the tree or isolate a single feature in the leaf node.
5) The final form is to reach an isolated forest of features.
6) Find the average path length h(d) for each feature in the isolation forest, where d is the dataset. Equation 1 is showed that.

$$c(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \quad (1)$$

Where C(n) denoted of the average of h(d) and n is the number of leaves, H(t) is the harmonic

number that calculated using ln(t) + (Euler's constant= 0.5772156649), and the anomaly score can be computed by equation 2:

$$s(d,n) = 2^{\frac{-E(h(d))}{c(n)}} \quad (2)$$

Where E(h(d)) is the average of all h(d).

7) If the value of S(x, n) is near to one, it indicates that the data is more probable to be anomalous; If S(x, n) is near to zero, it indicates normal data.

C. Cluster-based Local Outlier Factor (CBLOF)

CBLOF was suggested by He, Xu, and Deng [29]. It describes anomalies as a result of local distances to neighboring clusters and the overall size of the specific clusters where the data point belongs.
It first divides data points into large and small clusters. Data points within a small cluster close to a nearby larger cluster are recognized as outliers. The local outliers could not represent a single point, instead being a tiny group of separated points. CBLOF considers both the distance between a data point and the closest cluster as well as the size of the cluster to which a data point belongs.CBLOF has several advantages [30]:

- It works well to identify outliers in data that tends to naturally cluster. It separates the data into clusters and classifies outliers according to how much they deviate from the cluster center or how well they fit into smaller clusters.
- CBLOF can manage heterogeneous datasets with clusters of varying sizes and shapes, making it helpful in real-world applications where data is often non-uniform.
- The outcomes generated by CBLOF are relatively simple to read, as one can understand why a specific point is an outlier due to its relationship to its cluster and the number of points in it. This is a significant benefit over other algorithms, whose findings can be difficult to interpret. The steps of CBLOF procedure are [31]:

1) A data point is given to exactly one cluster using K-means, which is a good clustering algorithm.

2) Clusters are ranked from large to small based on their size, and over time, data counts are calculated. The "large" clusters keep up to 90% of the data, while the "small" clusters keep the remaining 10%.

3) Finds a data point's distance towards the centroid and outlier score using two rules.Firstly,the distance between data points in a large cluster is measured from the cluster's centroid. The distance is multiplied by the number of data points in the cluster to determine the outlier score.The second rule,if a data point is in the smallest cluster, the distance is calculated using the centroid of the next large cluster. The calculation of the outlier score involves multiplying the distance by the amount of data present in the small cluster containing the corresponding data point.

5. Camel Herd Algorithm (CHA)

It is an optimal intelligent algorithm that relies on the collective behavior of camels' herd in the desert to solve complex problems. Its goal is to reach various solutions by exploring multiple paths and starting from different points. It also avoids falling into local optimum and reaching the global one. Camels form vast herds that can number up to a thousand. Each herd is headed by a leader whose purpose is to look for wetlands, water, and food using the humidity factor. They are recognized by their capacity to feel humidity from a distance[32].
The important parameters in this algorithm are the number of herds and the number of leaders in the herds, where each herd has a leader who guides it to find the optimal solution, in addition to the total number of camels in the herds and the humidity rate, which is randomly set at the beginning for each herd [33].
The basic idea behind how it operates is that, the herd is spread out in the space problem. The leader begins his task at a random starting point and spreads the rest of the camels to find neighbors with high humidity using the fitness function. The best neighbors are saved, and the procedure is repeated until the best solution is found. The camel herd procedure is revealed in Figure 1 [34]

Algorithm1: Pseudocode of Camel Herd Algorithm

```
Input: No. of camel (M), no. of herds (H), max_Humidity (maxH)
Output: best short path
Begin
For i = 1 to Hi Do
  //Choose leader (LHi )from the herd by using selection approach
End for
Repeat
      For i := 1 to Hi Do
      b := 1
      Initialize (Humidity)
      For j := 1 to length (LHi)
          For each solution Do
              Establish random neighbors (RN) of LHi // RN denote no. of
            camel except leader
          For z := 1 to RN Do
              (best neighbors) BNZ= BNZ * 1\ Humidity
               BNZ= LHi - BNZ \ dis (LHi ,BNZ )
        End for
        LHi [j] [b+1] = LHi [j] [b] + BNZ
        End for
       Update Humidity
       End for
    End for
Until achieve goal or maximum Humidity
End
```

Figure 1. Pseudo code of (CHA)

6. Description and Analysis of the Dataset

In this work, two datasets from the Kaggle website were used[35],[36]respectively. The first data set has 8000 items

and 21 features. Its features are real numbers except target class which is integer. The second dataset has 1048575 items and 21 important features.
Table II and Table III have an overview of the dataset's features for two dataset.

TABLE II. Features of the First dataset

No.	Feature	Explanation	Range per Liter
1	aluminum	Water is dangerous if higher than 2.8	0–5.05
2	ammonia	Water is dangerous if higher than 32.5	0.08–29.8
3	arsenic	Water is dangerous if higher than 0.01	0–1.05
4	barium	Water is dangerous if higher than 2	0–4.94
5	cadmium	Water is dangerous if higher than 0.005	0–0.13
6	chloramine	Water is dangerous if higher than 4	0–8.68
7	chromium	Water is dangerous if higher than 0.1	0–0.9
8	copper	Water is dangerous if higher than 1.3	0–2
9	flouride	Water is dangerous if higher than 1.5	0–1.5
10	bacteria	Water is dangerous if higher than 0	0–1
11	viruses	Water is dangerous if higher than 0	0–1
12	Lead	Water is dangerous if higher than 0.015	0–0.2
13	nitrates	Water is dangerous if higher than 10	0–19.8
14	nitrites	Water is dangerous if higher than 1	0–2.93
15	mercury	Water is dangerous if higher than 0.002	0- 0.1
16	perchlorate	Water is dangerous if higher than 56	0 – 60
17	radium	Water is dangerous if higher than 5	0–7.99
18	selenium	Water is dangerous if higher than 0.5	0 – 0.1
19	silver	Water is dangerous if higher than 0.1	0–0.5
20	uranium	Water is dangerous if higher than 0.3	0–0.9
21	is_safe	Target Class	not safe=0 , safe=1

TABLE III. Features of the Second dataset

No.	Feature	Maximum Concentration Limits
1	Turbidity	5 (10)**
2	pH	6.5-8.5*
3	Color	5 (15)**
4	Odor	Would not be objectionable
5	Total Dissolved Solids	1000
6	Conductivity	1500
7	Iron	0.3 (3)**
8	Manganese	0.2
9	Fluoride	0.5-1.5*
10	Lead	0.01
11	Chloride	250
12	Sulphate	250
13	Nitrate	50
14	Copper	1
15	Zinc	3
16	Chlorine	0.1-0.2*

Note:* These standards indicate the maximum and minimum limits.

** Figures in parenthesis are upper range of the standards recommended.

This study aimed to clarify the distribution of 20 and 21 features in first and second dataset respectively that utilized in water quality prediction. Figures 2 and 3 depict the various distributions of features after cleaning and deleting missing data.

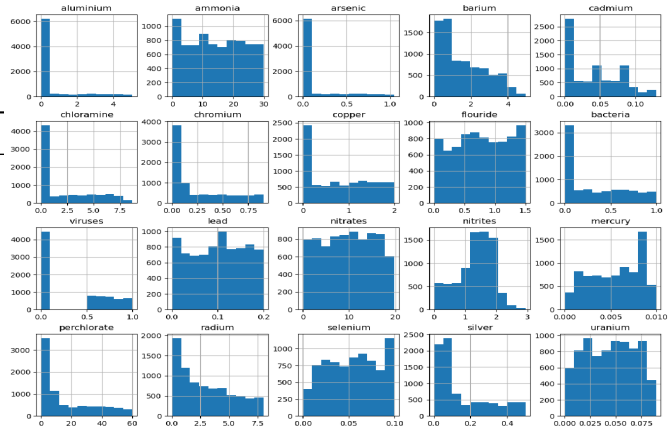

Figure 2. Distribution of water including chemicals in the First dataset

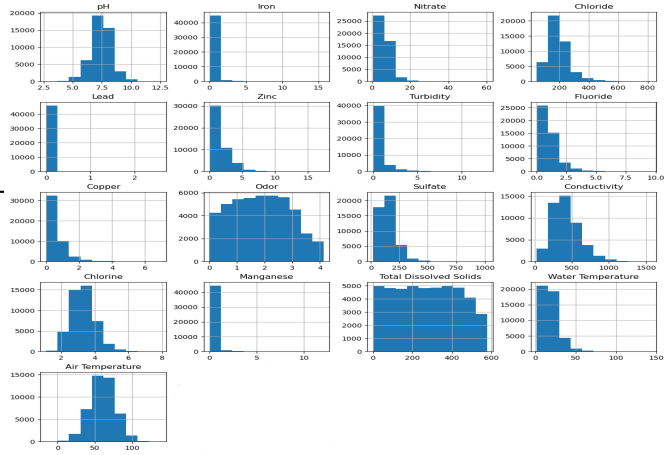

Figure 3. Distribution of water including chemicals in the second dataset

In addition, Table IV and Table V display a range of statistical values for input features in the two dataset. Also,it illustrated that the count of parameters in the first dataset is equal (7996.000000) but in the second was (46070.000000). In the first dataset minimum value was (-0.08000), which belongs to ammonia. perchlorate also achieved the maximum value and height standard deviation of (60.010000) and (17.688827) respectively. In the second data set, it was noted that the water temperature element had the minimum value with (-15.297819). However, Conductivity had the maximum value and height standard deviation with (1562.278417) and (187.240101) respectively.

7. Correlation Analysis (CA)

A correlation matrix is a table showing the correlation coefficients for numerous features. Every cell in the table

TABLE IV. Statistical Metrics on First Dataset

Material	count	mean	Standard deviation	min	25%	50%	75%	max
aluminium	7996.000000	0.666396	1.265323	0.0000	0.040000	0.070000	0.280000	5.050000
ammonia	7996.000000	14.278212	8.878930	-0.08000	6.577500	14.130000	22.132500	29.840000
arsenic	7996.000000	0.181477	0.252832	0.00000	0.030000	0.050000	0.100000	1.050000
barium	7996.000000	1.567928	1.216227	0.00000	0.560000	1.190000	2.482500	4.940000
cadmium	7996.000000	0.042803	0.036049	0.0000	0.008000	0.040000	0.070000	0.130000
chloramine	7996.000000	2.177589	2.567210	0.000000	0.100000	0.530000	4.240000	8.680000
chromium	7996.000000	0.247300	0.270663	0.000000	0.050000	0.000000	0.440000	0.900000
copper	7996.000000	0.805940	0.653595	0.000000	0.090000	0.750000	1.390000	2.000000
flouride	7996.000000	0 0.771648	0.435423	0.000000	0.407500	0.770000	1.160000	1.500000
bacteria	7996.000000	0.319714	0.329497	0.000000	0.000000	0.220000	0.610000	1.000000
viruses	7996.000000	0 0.328706	0.378113	0.000000	0.002000	0.008000	0.700000	1.000000
lead	7996.000000	0.099431	0.058169	0.000000	0.048000	0.102000	0.151000	0.200000
nitrates	7996.000000	9.819250	5.541977	0.000000	5.000000	9.930000	14.610000	19.830000
nitrites	7996.000000	1.329846	0.573271	0.000000	1.000000	1.420000	1.760000	2.930000
mercury	7996.000000	0.005193	0.002967	0.000000	0.003000	0.005000	0.008000	0.010000
perchlorate	7996.000000	16.465266	17.688827	0.000000	2.170000	7.745000	29.487500	60.010000
radium	7996.000000	2.920106	2.322805	0.000000	0.820000	2.410000	4.670000	7.990000
selenium	7996.000000	0.049684	0.028773	0.000000	0.020000	0.050000	0.070000	0.100000
silver	7996.000000	0.147811	0.143569	0.000000	0.040000	0.080000	0.240000	0.500000
Uranium	7996.000000	0.044672	0.026906	0.000000	0.020000	0.050000	0.070000	0.090000

TABLE V. Statistical Metrics on Second Dataset

Material	count	mean	Standard deviation	min	25%	50%	75%	max
pH	46070.000000	7.429783	0.899168	2.720970	6.859859	7.431281	8.013138	12.508936
Iron	46070.000000	0.163643	0.504624	0.000000	0.000074	0.006964	0.094816	15.748603
Nitrate	46070.000000	6.455556	3.277018	0.600649	4.240407	5.891040	7.961255	60.373341
Chloride	46070.000000	191.162543	73.212160	43.115394	141.197959	180.053353	225.681904	821.340029
Lead	46070.000000	0.002480	0.040075	0.000000	0.000000	0.000000	0.000000	2.577206
Zinc	46070.000000	1.591325	1.509082	0.000047	0.498047	1.168112	2.252260	17.456025
Turbidity	46070.000000	0.630641	0.992825	0.000000	0.065759	0.283541	0.767409	12.860362
Fluoride	46070.000000	1.024760	0.824096	0.000639	0.435440	0.842447	1.405219	9.595659
Copper	46070.000000	0.574632	0.608193	0.000002	0.167000	0.407499	0.773385	6.935948
Odor	46070.000000	1.873636	1.039409	0.011032	1.008938	1.864526	2.691711	4.140538
Sulfate	46070.000000	150.295230	73.055630	18.485174	98.993197	136.187879	187.047766	1021.964684
Conductivity	46070.000000	421.515282	187.240101	24.259860	284.770387	394.881144	527.529420	1562.278417
Chlorine	46070.000000	3.305040	0.721968	1.108282	2.803668	3.252339	3.741812	7.891530
Manganese	46070.000000	0.147796	0.501985	0.000000	0.000021	0.002950	0.043187	12.116501
Total Dissolved Solids	46070.000000	274.250508	159.969885	0.013855	136.094238	273.262287	408.969775	579.759551
Water Temperature	46070.000000	18.900567	10.504284	1.543869	11.766502	16.543181	23.325443	143.155393
Air Temperature	46070.000000	60.246895	17.100232	-15.297819	48.805324	60.312304	71.572667	137.632506
Air Temperature	46070.000000	60.246895	17.100232	-15.297819	48.805324	60.312304	71.572667	137.632506

represents a correlation between two value pairs [37]. The correlation matrix for two datasets with the output and each other is shown in Figure 4 and Figure 5.

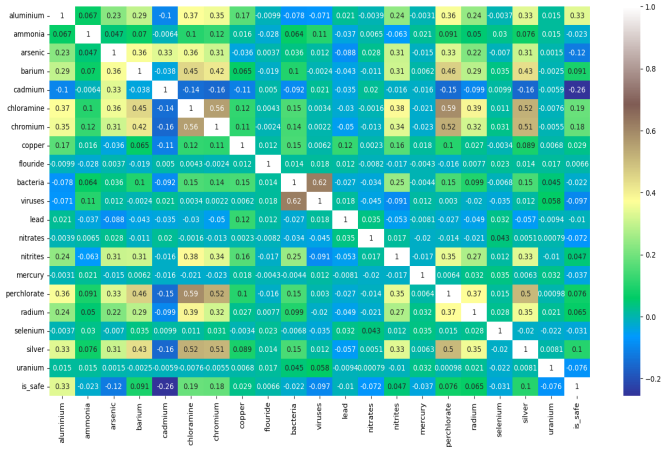

Figure 4. Correlation Matrix of First Dataset

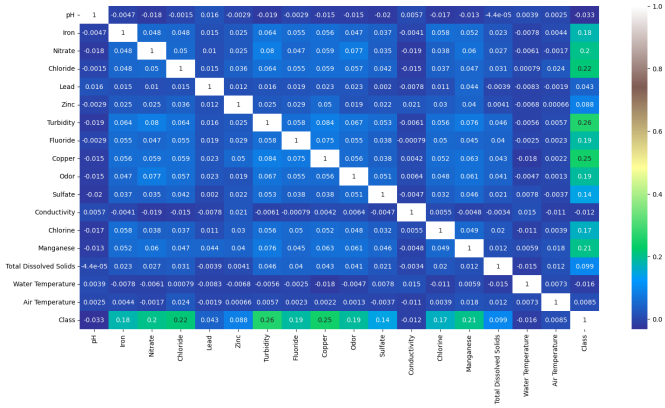

Figure 5. Correlation Matrix of Second Dataset

For example, the Figure 4 reveals that viruses has a positive correlation with bacteria (0.62), implying that increases in viruses cause increases in bacteria and vice versa. Silver has a positive correlation with the elements

chloramine and chromium (0.52) and (0.51), respectively, but it has a negative correlation with the element lead (-0.057), implying that as silver grows, lead slightly decreases.
On the other side, it was discovered that silver has a moderate correlation with perchlorate (0.50), implying that as silver levels rise, perchlorate will increase moderately.

8. Research Methodology and Approach

A. Research Requirement

1) Environmental Requirement.
   - windows OS.
   - Anaconda includes Jupyter notebook tools for the Python programming language.
2) Functional Requirement.

A group of libraries was used in the Python language to implement the desired goals, which are sk-learn, pandas, NumPy, matplotlib, lgbm, niapy and seaborn.

B. Proposed Methodology

The proposed methodology of the water quality prediction model consists of five phases: Preprocessing phase, Unsupervised ML phase, Tuning phase, Prediction using supervised ML phase and finally Performance evaluation phase. The framework of the proposed methodology can be simply described in Figure 6.

1) Pre-processing Phase Pre-processing is essential for improving the quality of data analysis. It refers to the act of acquiring and manipulating numerous data components in order to produce usable and relevant information. Pre-processing phase included Data cleaning, Data normalization, Data splitting, and lastly resampling training data.
   - Data Cleaning Cleaning data was performed by deleting records that contained incomplete data.
   - Data Normalization Normalization is a technique for standardizing attribute values in a dataset by placing data in a predefined range between 0 and 1 without affecting the underlying distribution. It ensures that the data keeps its original shape when scaled to a defined range. Equation (3) calculated the feature's normalization on a scale of 0 to 1 [38].

$$f_{scaled} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{3}$$

     Where f_{max} represents the feature's maximum value and f_{min} refers the minimum value. This is done by using MinMaxScaler function (f scaled).
   - Data Splitting The data was divided into two groups, with 70% going to training and 30% going to the testing technique.
   - Resampling Training Dataset Unbalanced datasets have unequal categories, one with more samples than the other. Classifiers may perform effectively in the majority class but poorly on the minority due to their greater effect. Unbalanced datasets often need to be resampled to achieve a more even distribution of class states [39],[40].
     SMOTE sampling, an adaptive oversampling approach, has been applied to process the raw dataset for guaranteeing high accuracy of the training data. The SMOTE approach efficiently motivates the minority class to become broader. Oversampling the minority class is a technique for dealing with unbalanced datasets. Duplicating samples in the minority class is the simplest solution, but these examples add no new information to the model [41].
2) Unsupervised ML Phase At this phase, three models of unsupervised ML were used, and each model utilized features without labels (outcome). The function of models was to discover anomaly score for each data point(outlier), which is added and fused with original dataset as an additional feature for using in the prediction algorithm by LGBM.
   Finally, a jointly hybrid model was proposed called (HLGBM+Fusion CIC) that combines LGBM and unsupervised ML(COPOD, IForest, CBLOF) after fusing their outliers. The hybrid model jointed the outcomes of fusion unsupervised algorithms (COPOD, IForest, and CBLOF) with original dataset and passed as input for the LGBM algorithm to achieve multi-model learning and highly representative prediction [42].
3) Tuning Hyperparameter Phase Choosing the correct hyperparameters has a significant impact on the effectiveness of the prediction model, and also allows for a more optimal solution with a better level of accuracy, but it is a difficult matter to achieve. So, swarm intelligent algorithms have demonstrated their capacity to perform such jobs.
   The camel herd algorithm was applied for tuning the hyperparameters of LGBM algorithm on two datasets, and the Table VI and Table VII showed the best one.

TABLE VI. Hyper-Parameter Models(First Dataset)

Models	Hyperparameters
LGBM	num_leaves=141, n_estimators=196
LGBM + COPOD	num_leaves=46, n_estimators=352
LGBM + IForest	num_leaves=44, n_estimators=333
LGBM + CBLOF	num_leaves=81, n_estimators=242
HLGBM+Fusion CIC	num_leaves=46, n_estimators=352

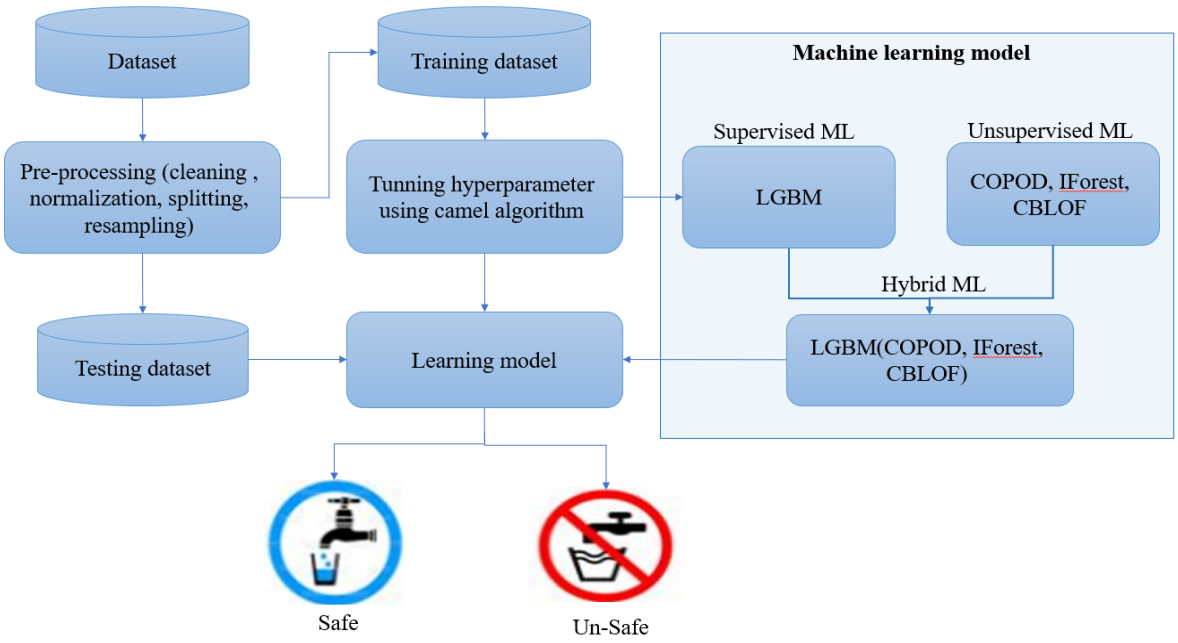

Figure 6. Methodology Framework

TABLE VII. Hyper-Parameter Models(Second Dataset)

Models	Hyperparameters
LGBM	num_leaves=228, n_estimators=294
LGBM + COPOD	num_leaves=244, n_estimators=328
LGBM + IForest	num_leaves=216, n_estimators=301
LGBM + CBLOF	num_leaves=239, n_estimators=239
HLGBM+Fusion CIC	num_leaves=113, n_estimators=391

In the above two tables, the hyper-parameters (n_leaves) and (n_estimator) control the number of leaves in a single tree and the number of trees in the model, respectively. These hyper-parameters are crucial for preventing overfitting and underfitting, as well as achieving high accuracy and performance. Increasing their value causes overfitting and complicates the model's structure, while decreasing their value,underfitting occurs, which is the model's inability to discriminate data throughout the training and testing phases.So, in this work Camel Herd Algorithm was used to create balanced values with great precision and simplicity, while preventing overfitting and underfitting .

4) Prediction using Supervised ML Phase At this stage, the water quality prediction process is carried out after pre-processing the dataset and tuning the hyperparameters of the LGBM algorithm.
5) Performance Evaluation After designing the model, its performance was evaluated using multiple metrics, including ROC AUC, precision, recall, f1 score, and accuracy. AUC-ROC is a classification metric that measures how effectively a classifier can distinguish between classes at different thresholds. AUC-ROC illustrates the trade-off within specificity and sensitivity in tests that produce numerical results rather than a binary positive or negative outcome. The AUC-ROC (decision thresholds) determines the optimum cut-off for both sensitivity and specificity. Accuracy represents categorization task performance and counts the number of accurately estimated examples across all data samples. Furthermore, Recall is an appropriate statistic for identifying model faults as well as how accurately the model recognizes actual "safe" and "non safe" occurrences. Precision refers to the percentage of positively (either "safe" or "non safe") identifies that have been correct. Precision measures quality, whereas recall measures quantity. F1 score is a statistic that aims to find a balance between precision and recall. These metrics are defined in 4,5,6,7 equations as follows :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

TP, FP, TN, and FN represent True Positive, False Positive, True Negative, and False Negative, respectively. They range from zero to one and used to determine the ML model that performs better to

identify ”safe” and ”nonsafe” instances [43],[44].

C. Result and Discussion

The results of LGBM model were evaluated before and after the SMOTE process for two datasets, as shown in Table VIII and Table IX.

TABLE VIII. Evaluation Metrics of LGBM Model(first dataset)

Evaluation metrics	Before SMOTE	After SMOTE
AUC	0.909	0.984
Precision	0.909	0.983
Recall	0.830	0.986
F1-score	0.867	0.985
Accuracy	0.971	0.984

TABLE IX. Evaluation Metrics of LGBM Model(Second Dataset)

Evaluation metrics	Before SMOTE	After SMOTE
AUC	0.862	0.908
Precision	0.656	0.869
Recall	0.868	0.965
F1-score	0.747	0.915
Accuracy	0.860	0.909

The previous tables show that the results of applying LGBM model on two datasets after SMOTE are better than before applying SMOTE, because balanced data ensures that the LGBM model is not biased or overfitted.
Moreover, for comparison with the traditional methods, Table X and Table XI show that LGBM algorithm outperformed them,like: SVM, NB (Naive Bayes), KNN, DT,and LR in terms of performance evaluation results and algorithm execution time for two datasets.
It was also observed that DT algorithm is the best among traditional algorithms due to its closeness to LGBM algorithm because the latter is based on DT algorithm. LGBM algorithm combines Gradient-Based Sampling (GOSS) and Exclusive Feature Pooling (EFB) where this combination gives high efficiency, accuracy and regression in data classification and reduces the cost of loss because it is based on dividing the tree into leaves and not on the depth level used in previous boosting algorithms.

TABLE X. comparison between LGBM with traditional ML(First Dataset)

Evaluation metrics	LGBM	SVM	NB	KNN	DT	LR
AUC	0.984	0.794	0.791	0.856	0.953	0.787
Precision	0.983	0.808	0.799	0.788	0.953	0.822
Recall	0.986	0.777	0.782	0.981	0.953	0.737
F1-score	0.985	0.792	0.791	0.874	0.953	0.777
Accuracy	0.983	0.794	0.791	0.857	0.953	0.787
Time	2.390625	2.96875	0.0	0.0	0.15625	0.015625

TABLE XI. comparison between LGBM with traditional ML(Second Dataset)

Evaluation metrics	LGBM	SVM	NB	KNN	DT	LR
AUC	0.908	0.675	0.700	0.763	0.860	0.748
Precision	0.869	0.787	0.845	0.727	0.861	0.766
Recall	0.965	0.486	0.494	0.856	0.863	0.721
F1-score	0.915	0.600	0.623	0.7864	0.862	0.743
Accuracy	0.909	0.673	0.697	0.764	0.860	0.747
Time	2.1718	84.5781	0.0312	6.875	2.2968	0.0625

Figure 7 displays confusion matrix for the LGBM model before and after SMOTE, and Figure 8 shows the height of AUC curve after oversampling in comparison with before for the first dataset.

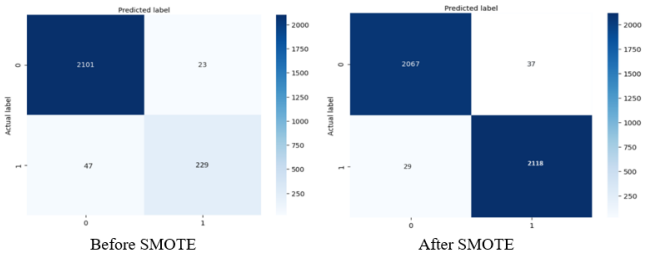

Figure 7. Confusion Matrix of LGBM Model(First Dataset)

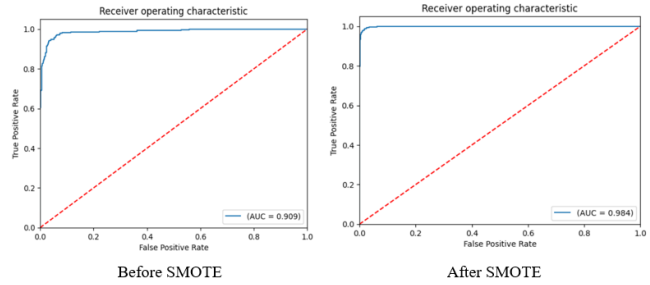

Figure 8. AUC of LGBM Model(First Dataset)

Figure 9 shows confusion matrix for the LGBM model before and after SMOTE, whereas Figure 10 shows the height of the AUC curve after SMOTE for the second dataset.

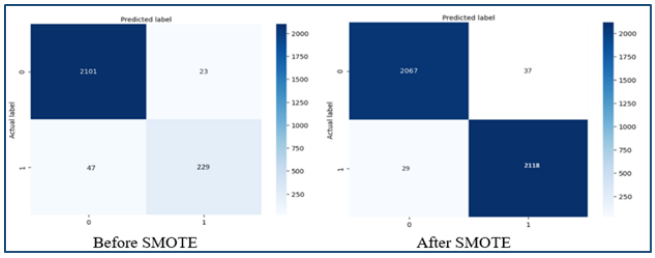

Figure 9. Confusion Matrix of LGBM Model(Second Dataset)

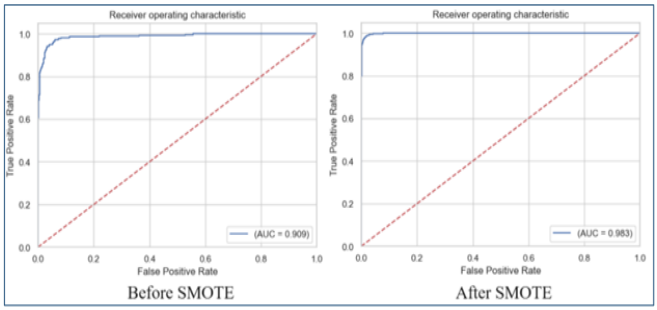

Figure 10. AUC of LGBM Model(Second Dataset)

After applying SMOTE algorithm on the original training dataset, tuning hyper parameters of LGBM algorithm was performed using Gamel herd algorithm. Next, LGBM algorithm was hybridized with the outliers generated by COPOD algorithm.
Similarly, the same procedure was repeated independently once on the IForest algorithm and once on the CBLOF algorithm. Finally, the results of three unsupervised algorithms (COPOD, IForest, and CBLOF) were fused as input to LGBM algorithm.
Table XII and Table XIII represent the performance evaluation results and execution time for all previous models, which show that (LGBM+IForest) model overcome (LGBM+COPOD) model and which show that (LGBM+CBLOF) model overcome (LGBM+IForest) model.
Finally proposed model (HLGBM+Fusion CIC) superior on the three previous models (COPOD, IForest, CBLOF).

TABLE XII. Performance Evaluation and Execution Time (First Dataset)

Evaluation metrics	LGBM Before tuning	LGBM After tuning	After Tuning Hybrid (LGBM+ COPOD)	Hybrid (LGBM+ IForest)	Hybrid (LGBM+ CBLOF)	HLGBM +Fusion CIC
AUC	0.984	0.986	0.987	0.989	0.990	0.992
Precision	0.983	0.986	0.987	0.990	0.989	0.990
Recall	0.986	0.986	0.988	0.990	0.991	0.993
F1-score	0.985	0.986	0.987	0.989	0.990	0.992
Accuracy	0.984	0.986	0.987	0.989	0.990	0.992
Time(Sec.)	2.390	7.687	11.890	12.921	10.359	8.265

TABLE XIII. Performance Evaluation and Execution Time (Second Dataset).

Evaluation metrics	LGBM Before tuning	LGBM After tuning	After Tuning Hybrid (LGBM+ COPOD)	Hybrid (LGBM+ IForest)	Hybrid (LGBM+ CBLOF)	HLGBM +Fusion CIC
AUC	0.908	0.910	0.911	0.917	0.913	0.922
Precision	0.869	0.873	0.874	0.883	0.875	0.894
Recall	0.965	0.964	0.966	0.965	0.969	0.961
F1-score	0.915	0.917	0.917	0.922	0.920	0.927
Accuracy	0.909	0.911	0.912	0.918	0.914	0.923
Time(Sec.)	2.765	34.203	40.640	37.687	45.406	27.734

Figures 11,12,13 depict the results of the confusion matrix for all applied models on the first dataset.
Figures 14,15,16 show the results of the confusion matrix for all applied models on the second dataset, demonstrating that the proposed model (HLGBM+Fusion CIC) overcomes the other models.

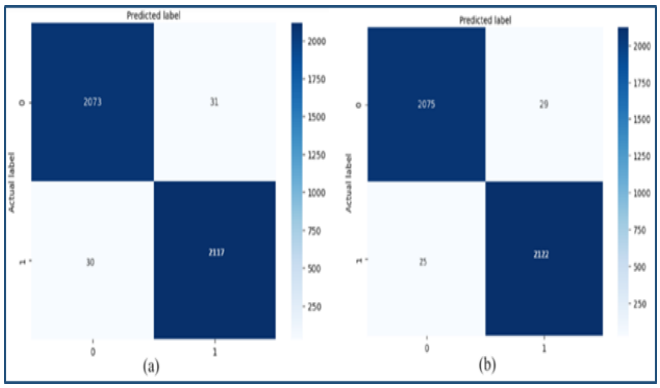

Figure 11. (a) Confusion Matrix of (a) LGBM After Tuning, and (b) LGBM + COPOD Model (First Dataset)

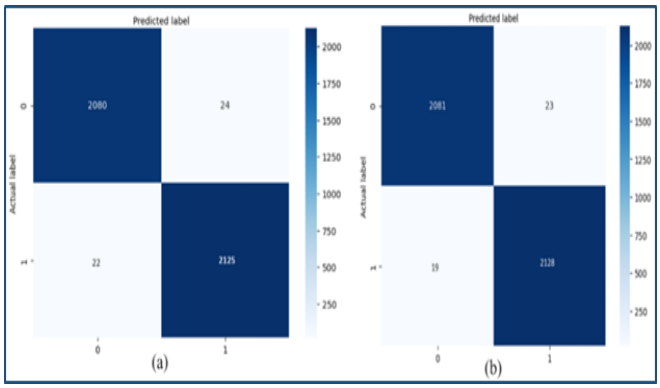

Figure 12. Confusion Matrix of (a) LGBM After Tuning + IForest, and (b) LGBM + COPOD Model(Second Dataset)

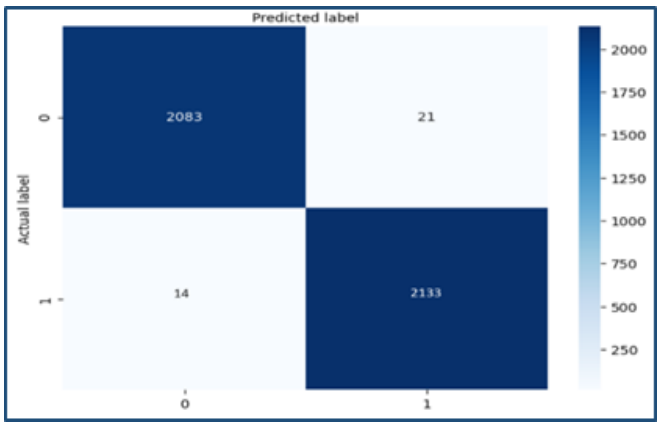

Figure 13. confusion matrix of (HLGBM+Fusion CIC) Model (First Dataset)

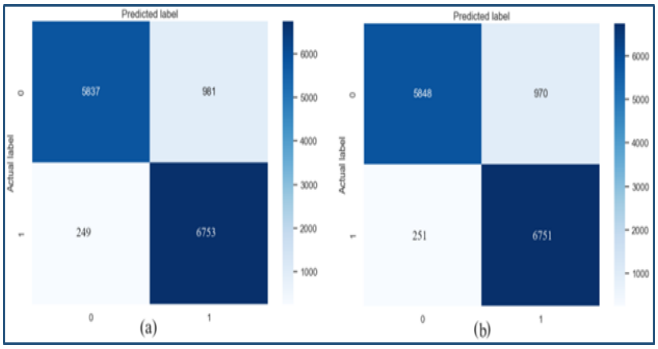

Figure 14. confusion matrix of (a) LGBM After tuning, and (b) LGBM After tuning + COPOD Model (Second Dataset)

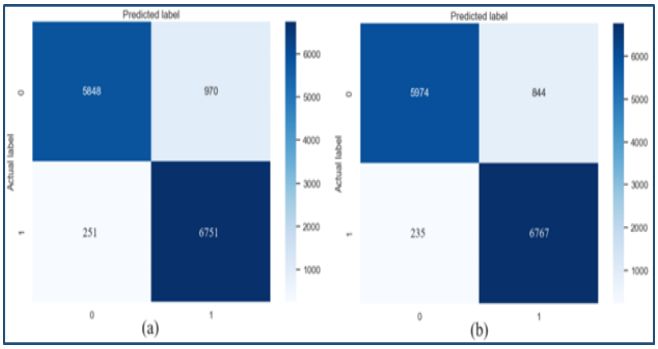

Figure 15. confusion matrix of (a) LGBM After tuning + IForest, and (b) LGBM After tuning + COPOD Model (Second Dataset)

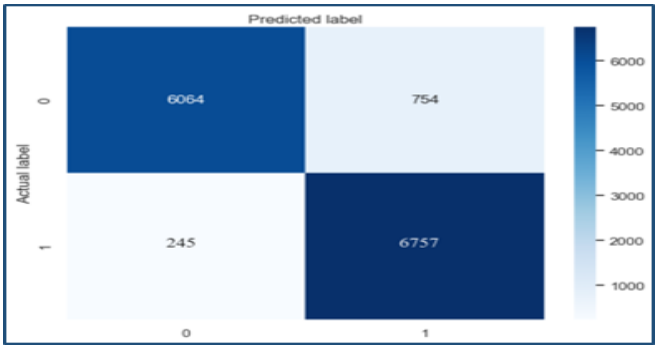

Figure 16. confusion matrix of (a) LGBM After tuning + IForest, and (b) LGBM After tuning + COPOD Model (Second Dataset)

The results of proposed model (HLGBM+Fusion CIC) were compared to the highest result mentioned in a related work conducted by Furqan Rustam et [10]. Table XIV shows that the result obtained from the proposed model was better compared for the previous study mentioned above.

TABLE XIV. Comparison with Related Work[10]

Paper	Accuracy	Precision	Recall	F1 score
Furqan Rustam et. [10]	0.96	0.91	0.87	0.89
HLGBM+Fusion CIC	**0.99**	**0.99**	**0.99**	**0.99**

9. Conclusion

Precise monitoring of changes in water quality is crucial for delivering drinking water. Conventional techniques like computing water quality index (WQI) can be time intensive and prone to mistakes. The global issues of water scarcity and pollution underscore the need to automate water suitability assessments. AI presents opportunities, for enhancing the analysis and forecasting of water quality.AI approaches can cut down expenses, help ensure adherence to water quality regulations and establishing monitoring systems is essential, for sustainable friendly water resource management.

This study focused on predicting the quality of water whether it is suitable for use or not. In order to achieve this, an evaluation and comparison of different models of supervised and unsupervised ML models was applied. The LGBM technique was compared before and after applying the SMOTE process. In addition, a comparison was made for hybrid models between supervised learning and unsupervised learning (COPOD, IForest, and CBLOF) after using SMOTE process and using swarm optimization to develop the model and get the best prediction outcomes.

The results show that the model (HLGBM+Fusion CIC) after SMOTE process ,adjusting the training dataset and jointing with fused unsupervised algorithms (COPOD, IForest, and CBLOF) and hybridizing it with LGMB algorithm outperforms other models, as an accuracy is 99% on first dataset and 92% on second dataset. Furthermore, these results have important implications for learning how to develop a new model that combines the features (outliers) of unsupervised ML with training dataset and then passed to supervised ML to achieve multi-model learning and highly representative prediction, including whether it is suitable for human consumption, agricultural irrigation, or other industrial or environmental applications.

References

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

[2] J. Wolfram, S. Stehle, S. Bub, L. L. Petschick, and R. Schulz, "Water quality and ecological risks in european surface waters–monitoring improves while water quality decreases," *Environment International*, vol. 152, p. 106479, 2021.

[3] N. Kedia, "Water quality monitoring for rural areas-a sensor cloud based economical project," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2015, pp. 50–54.

[4] K. Abirami, P. C. Radhakrishna, and M. A. Venkatesan, "Water quality analysis and prediction using machine learning," in *2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2023, pp. 241–245.

[5] O. Alshaltone, N. Nasir, F. Barneih, E. A. Majali, and A. Al-Shammaa, "Multi sensing platform for real time water monitoring using electromagnetic sensor," in *2021 14th international conference on developments in eSystems engineering (DeSE)*. IEEE, 2021, pp. 174–179.

[6] A. Y. Sun and B. R. Scanlon, "How can big data and machine learning benefit environment and water management: a survey of methods, applications, and future directions," *Environmental Research Letters*, vol. 14, no. 7, p. 073001, 2019.

[7] N. Nasir, O. Al Bashier, A. A. Murad, and M. Al Ahmad, "Optical detection of dissolved solids in water samples," in *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. IEEE, 2018, pp. 1–6.

[8] N. Nasir, M. Al Ahmad, and A. A. Murad, "Capacitive detection and quantification of water suspended solids," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*. IEEE, 2019, pp. 1–5.

[9] M. Ehteram, S. Ghotbi, O. Kisi, A. Najah Ahmed, G. Hayder, C. Ming Fai, M. Krishnan, H. Abdulmohsin Afan, and A. EL-Shafie, "Investigation on the potential to integrate different artificial intelligence models with metaheuristic algorithms for improving river suspended sediment predictions," *Applied Sciences*, vol. 9, no. 19, p. 4149, 2019.

[10] F. Rustam, A. Ishaq, S. T. Kokab, I. de la Torre Diez, J. L. V. Mazón, C. L. Rodríguez, and I. Ashraf, "An artificial neural network model for water quality and water consumption prediction," *Water*, vol. 14, no. 21, p. 3359, 2022.

[11] N. Nasir, A. Kansal, O. Alshaltone, F. Barneih, M. Sameer, A. Shanableh, and A. Al-Shamma'a, "Water quality classification using machine learning algorithms," *Journal of Water Process Engineering*, vol. 48, p. 102920, 2022.

[12] 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, 2020.

[13] M. Torky, A. Bakhiet, M. Bakrey, A. A. Ismail, and A. I. E. Seddawy, "Recognizing safe drinking water and predicting water quality index using machine learning framework," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023.

[14] F. Muharemi, D. Logofătu, and F. Leon, "Machine learning approaches for anomaly detection of water quality on a real-world data set," *Journal of information and telecommunication*, vol. 3, no. 3, pp. 294–307, 2019.

[15] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, "Efficient water quality prediction using supervised machine learning," *Water*, vol. 11, no. 11, p. 2210, 2019.

[16] M. M. Hassan, M. M. Hassan, L. Akter, M. M. Rahman, S. Zaman, K. M. Hasib, N. Jahan, R. N. Smrity, J. Farhana, M. Raihan *et al.*, "Efficient prediction of water quality index (wqi) using machine learning algorithms," *Human-Centric Intelligent Systems*, vol. 1, no. 3, pp. 86–97, 2021.

[17] M. Hmoud Al-Adhaileh and F. Waselallah Alsaade, "Modelling and prediction of water quality by using artificial intelligence," *Sustainability*, vol. 13, no. 8, p. 4259, 2021.

[18] S. Kouadri, A. Elbeltagi, A. R. M. T. Islam, and S. Kateb, "Performance of machine learning methods in predicting water quality index based on irregular data set: application on illizi region (algerian southeast)," *Applied Water Science*, vol. 11, no. 12, p. 190, 2021.

[19] A. Juna, M. Umer, S. Sadiq, H. Karamti, A. Eshmawi, A. Mohamed, and I. Ashraf, "Water quality prediction using knn imputer and multilayer perceptron," *Water*, vol. 14, no. 17, p. 2592, 2022.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[21] A. Shehadeh, O. Alshboul, R. E. Al Mamlook, and O. Hamedat, "Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, lightgbm, and xgboost regression," *Automation in Construction*, vol. 129, p. 103827, 2021.

[22] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, "Diagnosis of diabetes mellitus using gradient boosting machine (lightgbm)," *Diagnostics*, vol. 11, no. 9, p. 1714, 2021.

[23] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: copula-based outlier detection," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 1118–1123.

[24] R. K. Kennedy, Z. Salekshahrezaee, F. Villanustre, and T. M. Khoshgoftaar, "Iterative cleaning and learning of big highly-imbalanced fraud data using unsupervised learning," *Journal of Big Data*, vol. 10, no. 1, p. 106, 2023.

[25] X. Sun, Y. Wang, and Z. Shi, "Insider threat detection using an unsupervised learning method: Copod," in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 2021, pp. 749–754.

[26] W. Zhang and H. Fan, "Application of isolated forest algorithm in deep learning change detection of high resolution remote sensing image," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. IEEE, 2020, pp. 753–756.

[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[28] D. Xu, Y. Wang, Y. Meng, and Z. Zhang, "An improved data anomaly detection method based on isolation forest," in *2017 10th international symposium on computational intelligence and design (ISCID)*, vol. 2. IEEE, 2017, pp. 287–291.

[29] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[30] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

[31] P. Kasture and J. Gadge, "Cluster based outlier detection," *International Journal of Computer Applications*, vol. 58, no. 10, 2012.

[32] Z. O. Ahmed, A. T. Sadiq, and H. S. Abdullah, "Solving the traveling salesman's problem using camels herd algorithm," in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*. IEEE, 2019, pp. 1–5.

[33] A. K. Ardabili, Z. O. Ahmed, and A. L. Abbood, "Solving routing problem using improved camel herds algorithm," *International*

Journal on Perceptive and Cognitive Computing, vol. 6, no. 2, pp. 53–59, 2020.

[34] A. T. S. Al-Obaidi, H. S. Abdullah *et al.*, "Camel herds algorithm: A new swarm intelligent algorithm to solve optimization problems," *International Journal on Perceptive and Cognitive Computing*, vol. 3, no. 1, 2017.

[35] A. Kadiwal, "Kaggle dataset," Accessed: Feb. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/mssmartypants/water-quality/data.

[36] Vanthanadevis, "Kaggle dataset," Accessed: Aug. 9, 2024. [Online]. Available: https://www.kaggle.com/datasets/vanthanadevi08/water-quality-prediction/data.

[37] N. J. Gogtay and U. M. Thatte, "Principles of correlation analysis," *Journal of the Association of Physicians of India*, vol. 65, no. 3, pp. 78–81, 2017.

[38] H. A. Aldabagh and G. A. Altalib, "A survey for using ai techniques for predicting covid-19," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*. IEEE, 2022, pp. 1–63.

[39] A. Gökhan, C. O. Güzeller, and M. T. Eser, "The effect of the normalization method used in different sample sizes on the success of artificial neural network model," *International Journal of Assessment Tools in Education*, vol. 6, no. 2, pp. 170–192, 2019.

[40] H. A. Aldabagh and G. A. Altalib, "Predicting the effect of covid-19 on physical activity of survivors using gso and hybrid intelligent model," in *2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)*. IEEE, 2022, pp. 739–745.

[41] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[42] R. T. Ibrahim and F. M. Ramo, "Hybrid intelligent technique with deep learning to classify personality traits," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 231–244, 2023.

[43] J. Tohka and M. Van Gils, "Evaluation of machine learning algorithms for health and wellness applications: A tutorial," *Computers in Biology and Medicine*, vol. 132, p. 104324, 2021.

[44] R. T. Ibrahim and F. M. Ramo, "Classification of personality traits by using pretrained deep learning models," in *2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM)*. IEEE, 2021, pp. 42–47.