Journal of Water Process Engineering 48 (2022) 102920

Contents lists available at ScienceDirect

Journal of Water Process Engineering

journal homepage: www.elsevier.com/locate/jwpe

Water quality classification using machine learning algorithms

Nida Nasir [a,*], Afreen Kansal [b], Omar Alshaltone [a], Feras Barneih [a], Mustafa Sameer [c], Abdallah Shanableh [a], Ahmed Al-Shamma'a [a]

[a] Research Institute of Science and Engineering, University of Sharjah, United Arab Emirates
[b] Department of Statistics, London School of Economics, United Kingdom
[c] National Institute of Technology, Patna, India

ARTICLE INFO

Keywords:
Water quality
Water quality index
Machine learning
Stack modelling
Meta classifier
Ensemble models

ABSTRACT

Monitoring water quality is essential for protecting human health and the environment and controlling water quality. Artificial Intelligence (AI) offers significant opportunities to help improve the classification and prediction of water quality (WQ). In this study, various AI algorithms are assessed to handle WQ data collected over an extended period and develop a dependable approach for forecasting water quality as accurately as possible. Specifically, various machine learning classifiers and their stacking ensemble models were used to classify the WQ data via the Water Quality Index (WQI). The studied classifiers included Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), CATBoost, XGBoost, and Multilayer Perceptron (MLP). The dataset used in the study included 1679 samples and their meta-data collected over nine years. In addition, precision-recall curves and Receiver Operating Characteristic curves (ROC) were used to assess the performance of the various classifiers. The findings revealed that the CATBoost model offered the most accurate classifier with a percentage of 94.51. Moreover, after applying stacking ensemble models with all classifiers, accuracy reached 100% in various Meta-classifiers. Furthermore, the CATBoost achieved the highest accuracy as a primary gradient boosting algorithm and a meta classifier. Therefore, the boosting algorithm is proposed as a reliable approach for the WQ classification. The analysis presented in this article presents a framework that can support the efforts of researchers working toward water quality improvement using artificial intelligence.

1. Introduction

Water composes more than two-thirds of the earth's surface and is a critical resource for living organisms. However, despite its abundance, the consumable form of water is limited [1]. Moreover, numerous ailments transmit through water; hence, real-time monitoring of water quality (WQ) is essential [2]. Commonly, assessing WQ entails collecting water samples from various sites at different time intervals and evaluating them in laboratories. However, manual sampling and laboratory analysis of WQ for any given water body or process can be inefficient, expensive and time consuming. As a result, intelligent systems are increasingly used to monitor WQ, especially when real-time data are needed [3,4].

Machine learning (ML) is a crucial part of artificial intelligence (AI) that allows a system to automatically learn and improve from experience without having to be explicitly programmed [5]. The techniques used in ML are based on a thorough examination of data to spot trends and update themselves accordingly [6,7]. ML offers great opportunities for assessing, classifying, and predicting WQ indicators in water studies. For example, ML models can effectively simulate hydrological processes and contaminants transport subject to the availability of adequate sets data [8]. Detecting WQ parameters benefits from the availability of sensors, such as photosensors that rely on determining the wavelength for a specific color [9]. For example, phosphorus detection can be achieved colorimetrically, with color resulting from a chemical reaction between a particular reagent and phosphorus. Other sensors rely on fluctuations in capacitance values, which can be used to detect various dissolved water contaminants [10]. Outputs of such techniques can generate a good amount of data processed using AI quickly, accurately, reliable, and accessible.

Mechanism-oriented models are multifunctional that can simulate WQ using generally complicated advanced systems data structures. Other models have been introduced for various aquatic systems. Examples include the WASP model [11] and QUAL model [12] for WQ in

* Corresponding author.
E-mail address: nnasir@sharjah.ac.ae (N. Nasir).

https://doi.org/10.1016/j.jwpe.2022.102920
Received 28 February 2022; Received in revised form 28 May 2022; Accepted 31 May 2022
Available online 18 June 2022

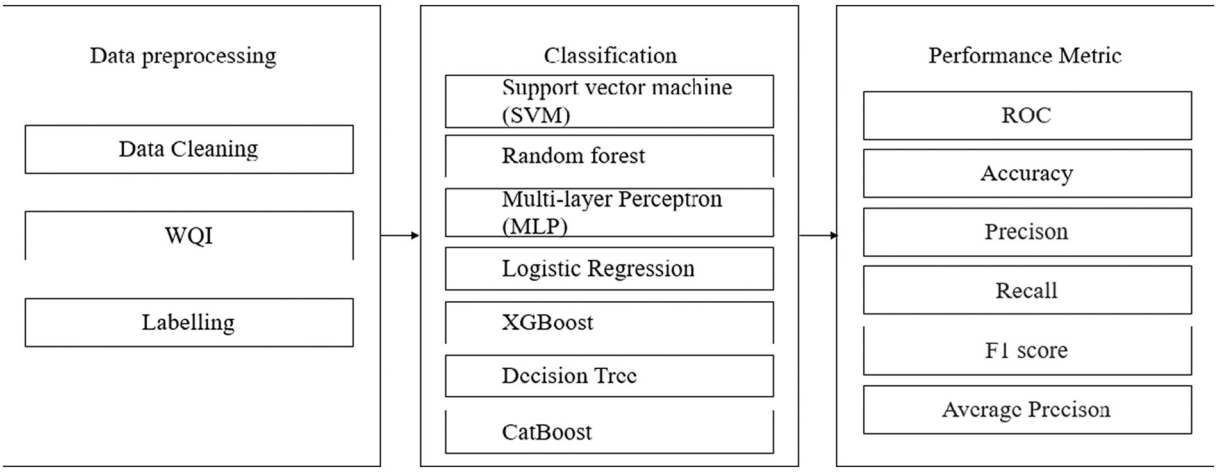


Fig. 1. Methodology of the proposed system.

Table 1
Standard values of clean water parameters [26].

Parameters	Permissible limits
Dissolved oxygen, mg/L	10
pH	8.5
Conductivity, _S/cm	1000
Nitrate, mg/L	45
Biological oxygen demand, mg/L	5
Fecal coliform/100 mL	100
Total coliform/100 mL	1000

Table 2
Water Quality Index (WQI) range classification [26,28].

WQI range	Classification
0–25 %	Clean
26–50 %	Unclean
56–75 %	Polluted
76–100 %	Highly polluted

rivers and the MIKE21 model [13] for seas and coastal waters. In addition, a two-dimensional (2D) numerical model based on the Environmental Fluid Dynamic Code (EFDC) was used to simulate the water environment in the Mudan River in northeast China [14]. In another approach based on satellite image fusion and mining techniques using Principal Component Analysis (PCA), the authors studied the surface WQ parameters in Lake Gala in Turkey [15].

Nowadays, researchers demonstrate the feasibility and effectiveness of artificial intelligence in WQ estimations. Liao and Sun [16] paired artificial neural networks (ANN) and decision tree algorithms to forecast WQ. Other researchers [17] proposed a deep learning network model that produced more accurate and better results than supervised learning-based techniques in forecasting dissolved oxygen and pH of the water. To classify WQ, Shafi et al. [18] used four ML algorithms: SVM, Neural Networks (NN), Deep Neural Networks (DNN), and KNN, in different water bodies, revealing that DNN surpassed other algorithms with an accuracy of 93 %. Gazzaz et al. [19] designed a feed-forward, fully connected, three-layer neural network model and the Internet of things (IoT) to calculate WQI in the Kinta River in Malaysia. Sakizadeh [20] used three different ANN algorithms, namely, ANN with early stopping, an ensemble of ANN, and ANN with Bayesian regularization, to predict WQI in 47 wells and springs in Iran the concentration of 16 WQ measurements. Abyaneh [21] estimated two WQ parameters in a wastewater treatment plant, namely, chemical oxygen demand (COD) and biochemical oxygen demand (BOD), using multivariate linear regression (MLR) and ANN. Liu et al. [22] proposed a model based on a Bi-directional Stacked Simple Recurrent Unit learning network (Bi-S-SRU) to estimate WQ for pH, water temperature, and dissolved oxygen in smart mariculture. Jaloree et al. [23] proposed a decision tree model to predict six quality parameters, namely ammonia-nitrogen (NH_3-N), nitrate-nitrogen (NO_3-N), pH, temperature, BOD, and COD, to estimate the WQ of the Narmada river in India. A comparative study of ANN algorithms was presented in [24] on estimating WQ for the Ganga river in India. The author used Levenberg Marquardt (LM) and gradient descent adaptive (GDA) to estimate WQ based on BOD, DO, and water temperature data. This paper is a step toward providing a detailed comparative study of machine learning classifiers to identify a model with the highest accuracy and most reliable results that can be extended for hardware implementation and other high-tech possibilities.

2. Methodology

Classification predictive modelling is the task of approximating a mapping function (type of classifier) from input variables (data) to discrete output variables (classification). Before employing ML models

Table 3
Statistical values for various input variables.

	Dissolved oxygen (mg/L)	pH	Conductivity (μmhos/cm)	B.O.D. (mg/L)	Nitrate N+ mg/L)	Fecal coliform (MPN/100 mL)	Total coliform (MPN/100 mL)	WQI
Count	1439	1439	1439	1439	1439	1180	1329	1179
Mean	6.61	7.18	537.72	3.13	1.84	264.84	536.69	68.14
Std	1.14	0.69	1313.88	6.04	2.73	379.97	672.35	36.14
Min	0.6	0	11	0.1	0	0	0	−54.25
25 %	6.28	6.9	75	1.1	0.26	20	69	50.34
50 %	6.8	7.2	143	1.725	0.57	127.5	256	61.28
75 %	7.2	7.6	332	3.2	2.4	322.25	701	78.59
Max	11.4	9.01	9753	88	20.45	2350	3000	450.38

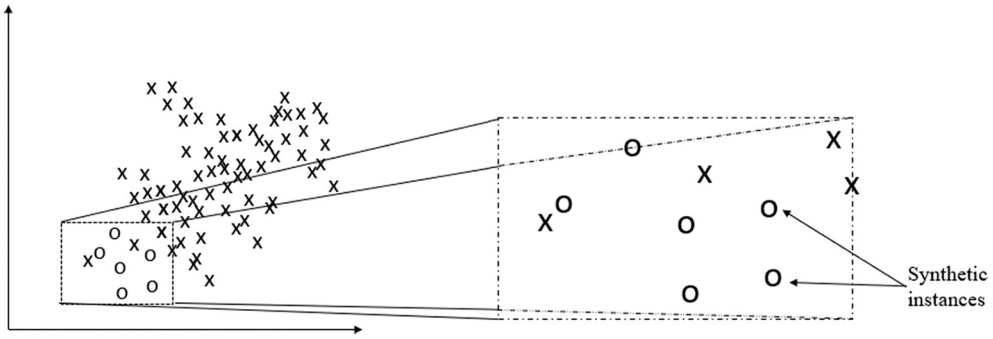


Fig. 2. Schematic diagram of the synthetic instances of the SMOTE algorithm.

Table 4
SVM default parameters.

Parameter	Value
Regularization parameter	1
Kernel	Radial basis function kernel
Degree	3
Gamma	'scale'
Coef0	0
Shrinking	True
Probability	False
Tolerance	0.001
Cache_size	200
Class_weight	None
verbose	False
Max_iter	−1
Decision_function_shape	'ovr'
Break_ties	False
random_state	None

Table 5
The default parameter of RF.

Parameter	Value
n_estimators	100
Criterion	'gini'
max_depth	None
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
max_features	'auto'
max_leaf_nodes	None
min_impurity_decrease	0
Bootstrap	True
oob_score	False
n_jobs	None
random_state	None
Verbose	0
warm_start	False
class_weight	None
ccp_alpha	0
max_samples	None

to analyze the data, prior actions were taken to prepare the data as input to the model. This included dividing the data into training and testing sets to train the 7 model and evaluate the performance. In addition, the dataset was cleaned by deleting incorrect values and replacing empty cells with the median of the dataset's input variables. Different ML models were then used to predict Water Quality Classification (WQC) using the identified factors (see Section 2.2 for all classifier details). The suggested technique for the current study is shown in Fig. 1. The following subsections explain each part of the framework of the proposed system and how it has been used.

Table 6
MLP parameters.

Parameter	Value
Hidden_layer_sizes	100
Activation	'relu'
Solver	'adam'
Alpha	0.0001
Batch_size	'auto'
learning_rate	'constant'
learning_rate_init	0.001
Power_t	0.5
max_iter	200
Shuffle	True
random_state	None
tolarance	0.0001
verbose	False
warm_start	False
Momentum	0.9
nesterovs_momentum	True
early_stopping	False
validation_fraction	0.1
Beta_1	0.9
Beta_2	0.999
Epsilon	1e-8
N_iter_no_change	10
max_fun	15,000

Table 7
Values of LR used in code.

Parameter	Value
Penalty	12
Dual	False
Tolerance	0.0001
C	1
Fit_intercept	True
IIntecept_scaling	1
Class_weight	None
Random_state	None
Solver	'lbfgs'
Max_iter	100
Multi_class	'auto'
Verbose	0
Warm_start	False
n_jobs	None
l1_ratio	None

2.1. Dataset description and processing

2.1.1. Dataset

This drinking water quality data was collected from various Indian states [25] between 2005 and 2014. A total of 1679 samples were collected and analyzed for dissolved oxygen (DO), pH, conductivity, biochemical oxygen demand (BOD), nitrate, fecal coliform, and total coliform. More information about the dataset can be obtained from the

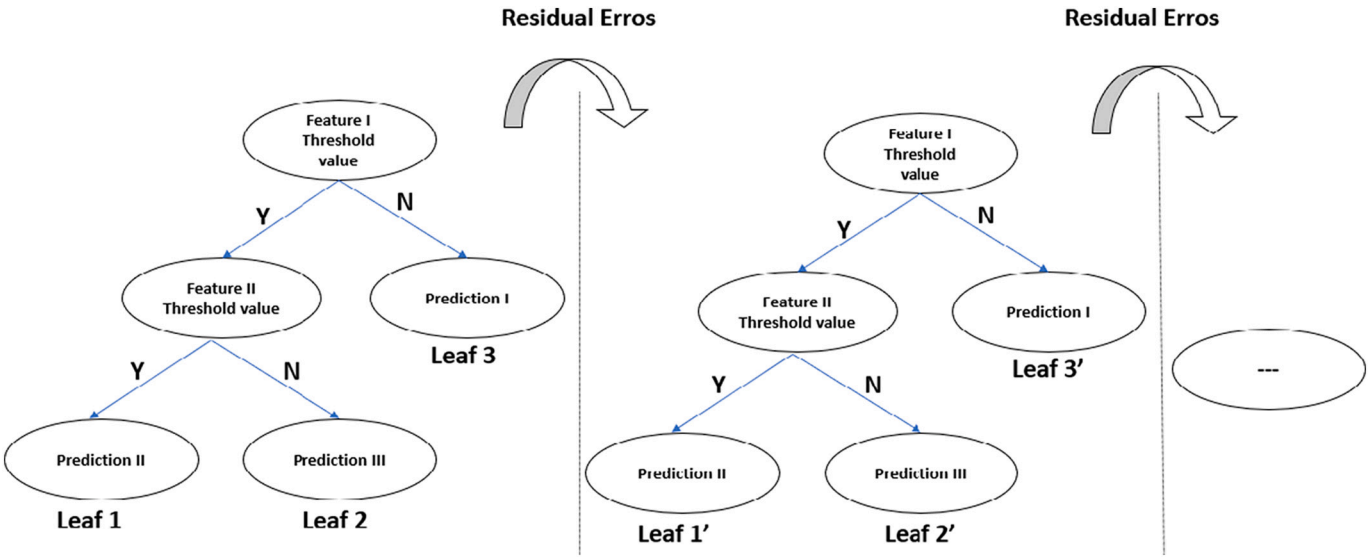


Fig. 3. Schematic diagram of XGBoost trees.

Table 8
Values of XGBoost.

Parameter	Value
Booster	gbtree
Verbosity	1
validate_parmeters	False
disable_ default_eval_metric	False

Table 9
Values of LR.

Parameter	Value
Criterion	'gini'
Splitter	'best'
max_depth	None
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
max_features	None
Random_state	None
max_leaf_nodes	None
Min_imputity_decrease	0
class_weight	None
Ccp_alpha	0

Table 10
The values of CATBoost used.

Parameter	Value
Iterations	10
learning_rate	0.1
Depth	2
loss_function	'MultiClass'

link mentioned in the data availability section. In addition, the data were subjected to a preprocessing step to improve data quality. Data preprocessing encompasses all steps required to avoid irrelevant entries by cleaning (i.e., removing outliers) and labeling the data.

2.1.2. Determination of water quality index

The water quality index (WQI) is calculated by averaging the individual index values of some or all the parameters used [26]. In this study, the WQI scores were used to categorize the water samples, and the following relation was used to compute the WQI:

$$WQI = \frac{\sum_{i=1}^{N} q_i \times w_i}{\sum_{i=1}^{N} w_i} \quad (1)$$

$$q_i = 100 \times \left(\frac{V_i - V_{Ideal}}{S_i - V_{Ideal}} \right) \quad (2)$$

$$w_i = \frac{k}{S_i} \quad (3)$$

$$K = \frac{1}{\sum_{i=1}^{N} S_i} \quad (4)$$

where N denotes the number of the total parameter, q_i denotes the quality estimate scale for each parameter i calculated by Eq. (2), w_i denotes each parameter's unit weight, V_i is a measured value that refers to the water samples tested, V_{Ideal} is an outstanding reference value and indicates pure water (0 for all parameters except DO = 14.6 mg/L and pH = 7.0), S_i is an expected value suggested for parameter i, as tabulated in Table 1 [26], and K in Eq. (4) denotes the proportionality constant. Water quality classification (WQC) is also a factor to consider. WQC may be calculated using the criteria shown in Table 2 [28]. In addition, Table 3 shows various statistical values (count, mean, standard deviation, min, 25 %, 50 %, 75 %, and max) for the dataset's input variables (e.g., DO, pH, dissolved oxygen, conductivity, BOD, Nitrate, and fecal

Table 11
Results of the proposed system with seven classifiers.

	Logistic regression	SVM	Decision tree	XGBoost	MLP	Random forest	CATBoost
Accuracy	0.7291	0.8068	0.81623	0.8807	0.8863	0.9393	0.9451
Precision	0.7247	0.81302	0.8169	0.8836	0.8890	0.9397	0.9458
Recall	0.7292	0.8068	0.8163	0.8807	0.8863	0.9393	0.9451
F1 score	0.7249	0.80601	0.8156	0.8804	0.8864	0.9394	0.9449

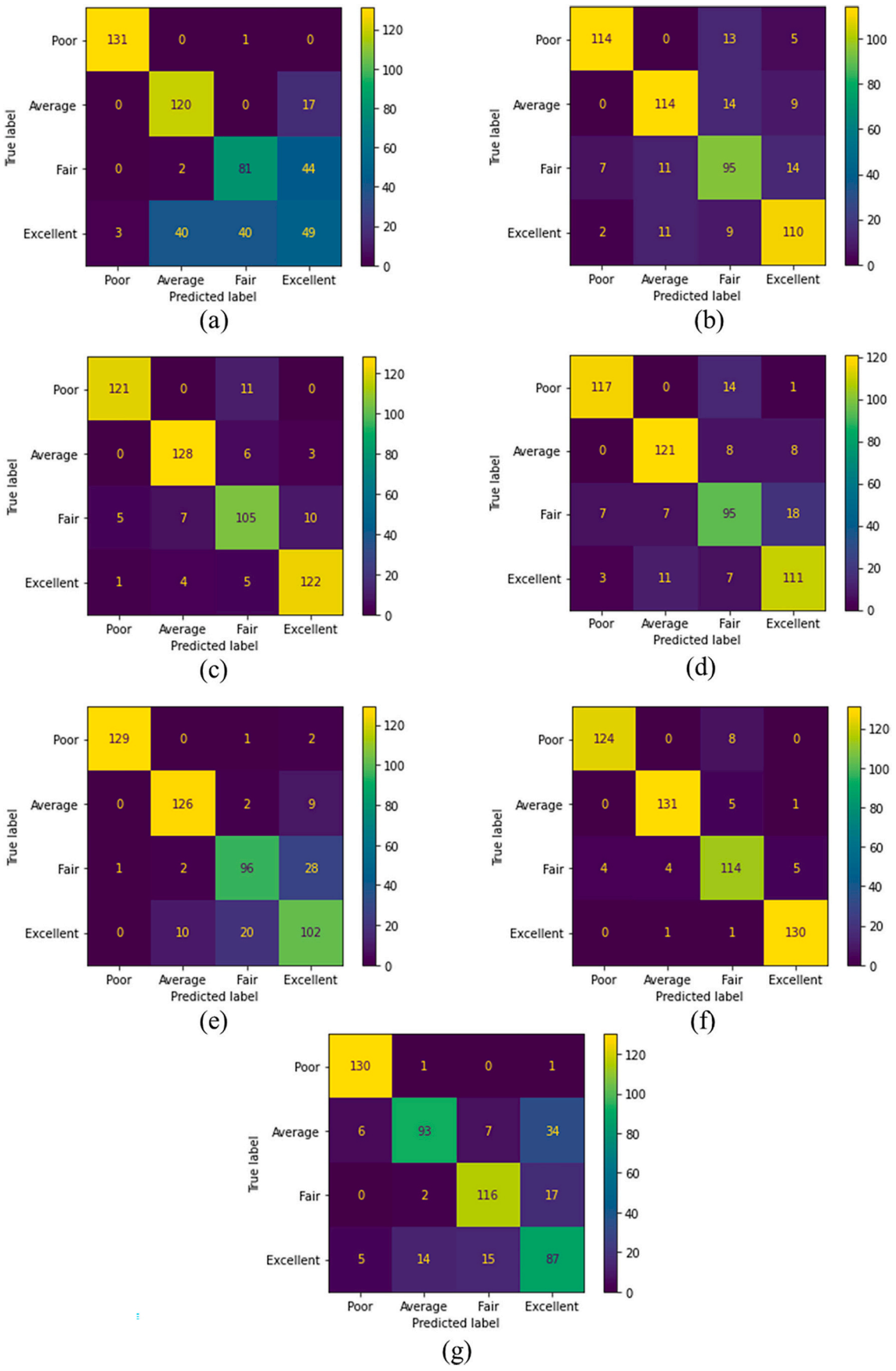


Fig. 4. Confusion matrix (Imputation with median and standard scaler) a) Logistic regression, b) Decision tree classifier c) Random Forest classifier d) XGB classifier e) MLP classifier f) CATBoost classifier g) SVM classifier.

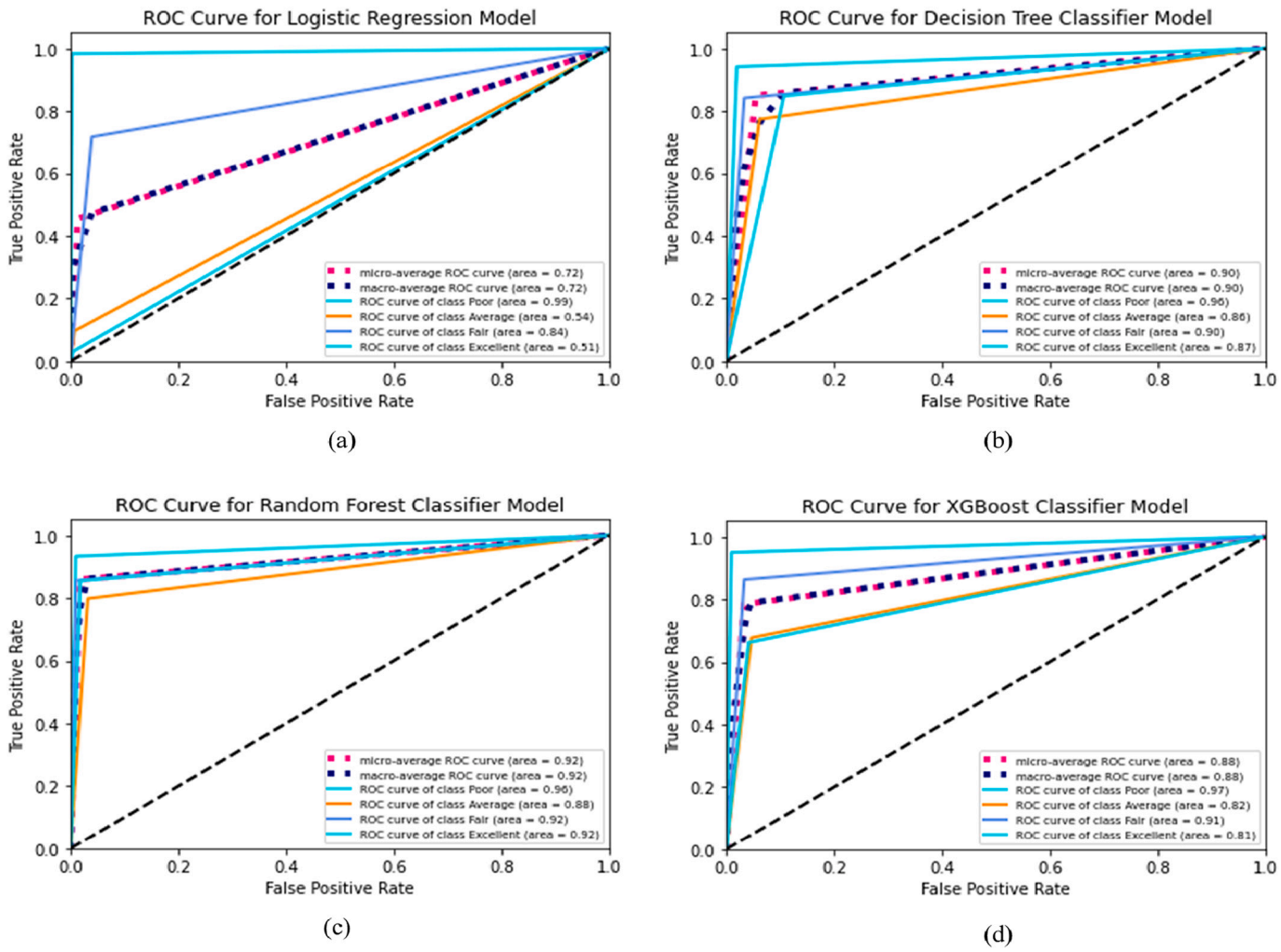


Fig. 5. ROC of a) Logistic regression, b) Decision tree classifier c) Random forest classifier d) XGB classifier e) MLP classifier f) Catboost classifier g) SVM classifier.

and total coliform) [27].

2.1.3. Data analysis: SMOTE

The synthetic minority oversampling technique (SMOTE) algorithm is an oversampling technique used for class imbalance. It adopts a subset of data from the minority classes and then produces related new synthetic examples. The synthetic instances are then added to the original data set. In this procedure, SMOTE creates a sample from the line between the minority class samples and their neighbors. The additional dataset may be utilized to train the classification model, thus overcoming overfitting produced by simple random oversampling [30]. The schematic diagram of the synthetic instances of the SMOTE algorithm is shown in Fig.2.

Repeated stratified k-fold cross-validation was used to evaluate the model, after defining the algorithm with any required hyperparameters (i.e., by using the defaults). Moreover, the three 10-fold cross-validation repeats were applied, which means the model can fit and evaluate 30 models on the dataset three times with 10-fold cross-validation. The dataset is stratified, which means that each fold of the cross-validation split has the same class distribution as the original dataset, with a 1:100 ratio in this case. On the other hand, the ROC area under curve (AUC) metric has been used to evaluate the model. This may be overly optimistic for severely imbalanced datasets, but it still shows a relative change with better-performing models.

2.2. Classifiers

For these simulations, Jupyter via Anaconda navigator was used with Python programming language. The machine used had an intel i7 core 7th generation processor and 16GB RAM. This section presents brief descriptions of all the classifiers used in the model.

2.2.1. Support Vector Machine (SVM)

The support vector machine (SVM) is a discriminative approach based on creating a hyper-plane to minimize errors that may be used for both classification and regression problems [31]. The observation dataset can be represented as follows:

$$D_s = \{(x_i, y_i)\}_{i=1}^{n} \tag{5}$$

where x_i represents the inputs and y_i defines the outputs with a linear function, as illustrated in Eq. (5) for modelling system (S) with observation dataset (D_s) that is calculated in Eq. (5) for numbers of variables n.

$$f(x) = \langle \omega^* \phi(x) + b \rangle \tag{6}$$

The maximum of Eq. (7) is the optimum function (subject to Eq. (8)). As SVM uses a new type of loss function. This means that the loss function can be employed in the form of an insensitive loss function.

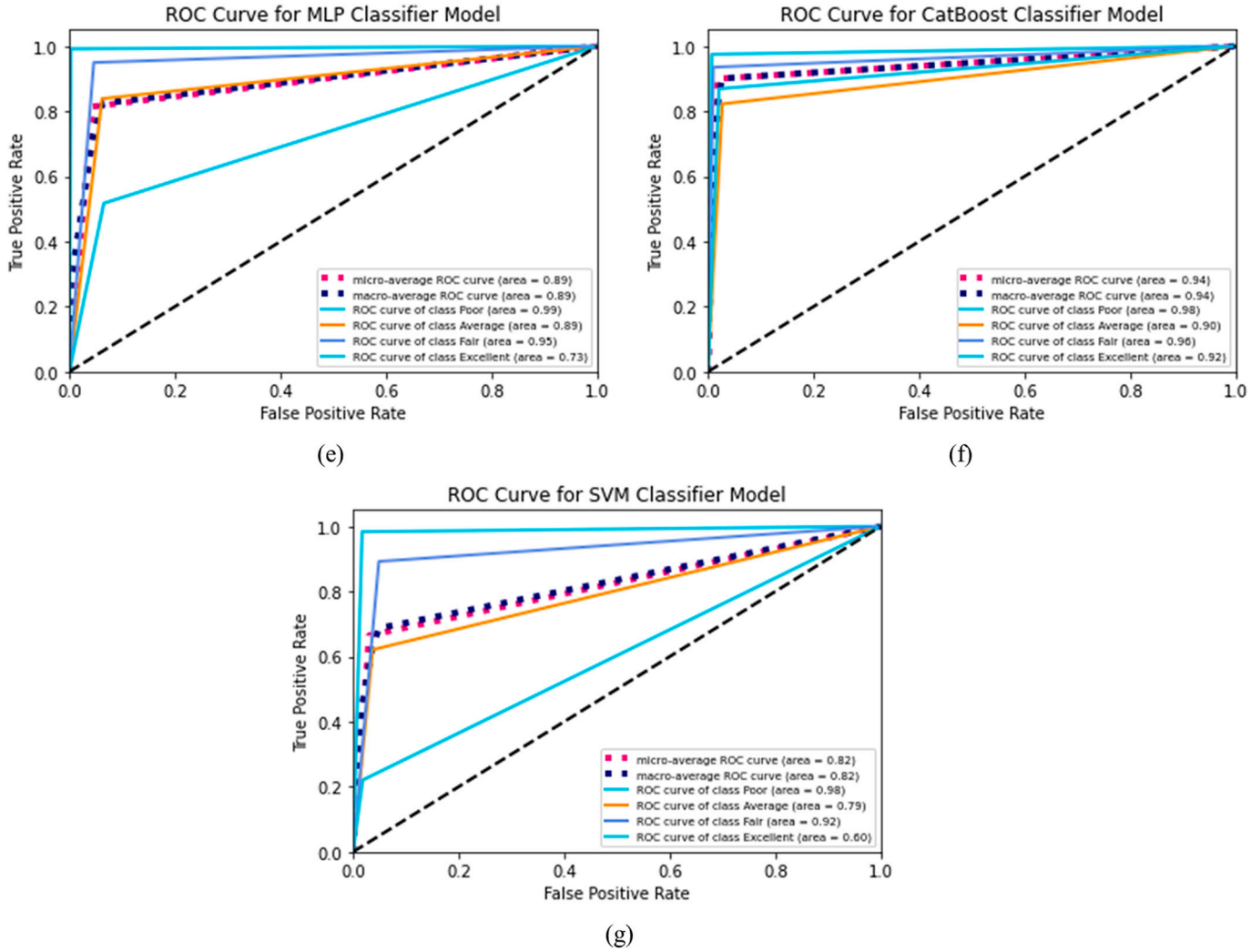


Fig. 5. (*continued*).

$$min(\omega, b, \xi^{-}, \xi^{+}) = \frac{1}{2}*\left|\left|\omega^{2}\right|\right| + C\sum_{i=1}^{n}(\xi i^{-} + \xi i^{+}) \tag{7}$$

$$S.t.\begin{cases} y_i - \omega^{T}*\phi(x) - b \le \varepsilon + \xi i^{-} \\ -y_i - \omega^{T}*\phi(x) + b \le \varepsilon + \xi i^{+} \\ \xi i^{-}, \xi i^{+} \ge 0 \\ i = 1, 2, \ldots n \end{cases} \tag{8}$$

Where $\phi(x)$ denotes a kernel function (k) as in a polynomial, radial basis, or linear function; b denote represents weigh and basis vectors; C denotes a pre-specified value to penalize the training error; ξ_i^{-} and ξ_i^{+} denote the lower and upper output constraints, respectively. Support vector machine is highly efficient in many cases, such as handling higher dimensional data when features are more than training examples and for inseparable classes when the almost negligible effect of outliers as hyperplane is impacted only by the support vectors. On the other hand, though SVM consumes more time and is unsuitable for a more extensive dataset, its performance gets low in case of overlapped classes; picking the proper kernel function can be tricky [32]. Moreover, the values of SVM used in the code are shown in Table 4.

2.2.2. Random forest

Regression and classification are conducted by the random forest (RF) technique that operates by constructing an ensemble of decision trees in training by swapping and changing the covariates to improve the prediction performance. The weighted average of tree outputs is used to achieve the goal, as RF has many potential features generated from different nodes. This model requires many trained trees and a certain amount of the variable in each tree. The RF classifier is a reliable algorithm that surpasses several other classification algorithms in terms of accuracy. The parameters that were utilized were as follows: estimators = 300, maximum depth = 100 and minimum sample split = 3 [33]. The first step for the RF classification technique is to randomly select K features from a list of m characteristics, where k << m; followed by finding d, using an ideal rift point, the node between the K features. Afterward, the node may be divided into two daughter nodes by using the most excellent rift, then repeat steps until the desired number of nodes has been reached; lastly, repetition of all steps results in the growth of n trees [34]. Additionally, the RF prediction process steps initially take the test characteristics and predict the outcome using the rules of each randomly generated decision tree, then save the expected result (target); after that, evaluate the votes for each potential target. Take the RF-algorithm final forecast as the highest chosen expected objective.

The values of RF used in code are shown in Table 5.

2.2.3. Multi-layer Perceptron (MLP)

In water modelling, the MLP is a popular ML model. An MLP model comprises three layers: input, hidden, and output. Neurons link these layers with weight and bias, weighted and biased [35]. Using an activation function (f), the weighted-variables are summed with the bias of

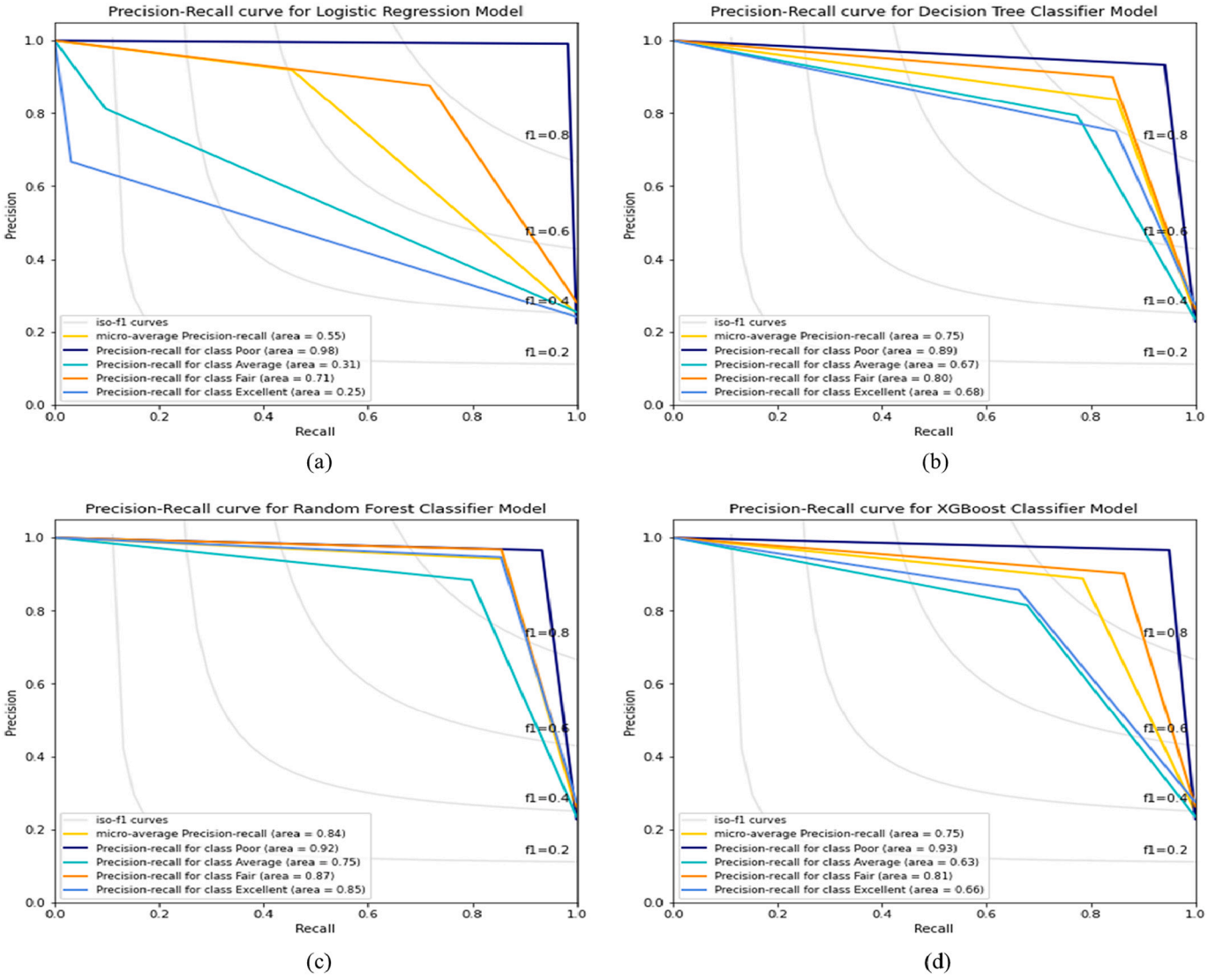


Fig. 6. Precision recall for: a) Logistic regression, b) Decision tree classifier c) Random forest classifier d) XGB classifier e) MLP classifier f) Catboost classifier g) SVM classifier.

the layer, which are converted from the jth layer to the jth + 1 layer; and so on, until the target layer. The training procedure is performed iteratively, adjusting the layer weights and biases until excellent preliminary performance (coefficient of correlation). The models will be used with three MLP layers to make things easier. The outputs Y_k will be provided by the following flowing equation.

$$Y_k = f_k\left(\sum_{i=1}^{m} W_{jk} * f_j\left(\sum_{i=1}^{n} X_i W_{ij}\right)\right) + W_O \tag{9}$$

The feature number is n, the hidden layer neuron numbers are m, the target layer neuron numbers are p, and the bias is W_0. The weights between the jth neuron and the kth target neuron and between the ith neuron and the jth neuron are W_{jk} accordingly. Whereas f_k and f_j are the transfer functions of the output and hidden layer neurons k and j, accordingly. In addition, the values of MLP used in code shown in Table 6.

2.2.4. Logistic regression

The logistic regression model (LRM) is the relation between a categorical dependent variable and a collection of independent (explanatory) factors, which can be used to estimate the probability of the occurrence of an event [36]. A mathematical model of a set of explanatory factors is used to predict the mean of a continuous dependent variable in multiple regression. The logit transformation is written mathematically, where p is the probability and the corresponding odds, as shown in Eq. (10). However, the values of LR used in the code are shown in Table 7.

$$l = \text{logit}(p) = ln\left(\frac{p}{1-p}\right) \tag{10}$$

2.2.5. XGBoost

Gradient boost machines (GBM) are among the top-performing algorithms for supervised learning, and XGBoost is one of the implementations of this method. In addition, it may be used to solve problems involving regression and classification [37]. XGBoost is desired by data scientists as it has a high execution speed out of core computation. The XGBoost operates as follows: Assume a dataset DS with m features and n examples$DS = \{(x_i, y_i); i = 1 \ldots n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$. Let y_i be the predicted output of an ensemble tree model derived from the Eq. (11):

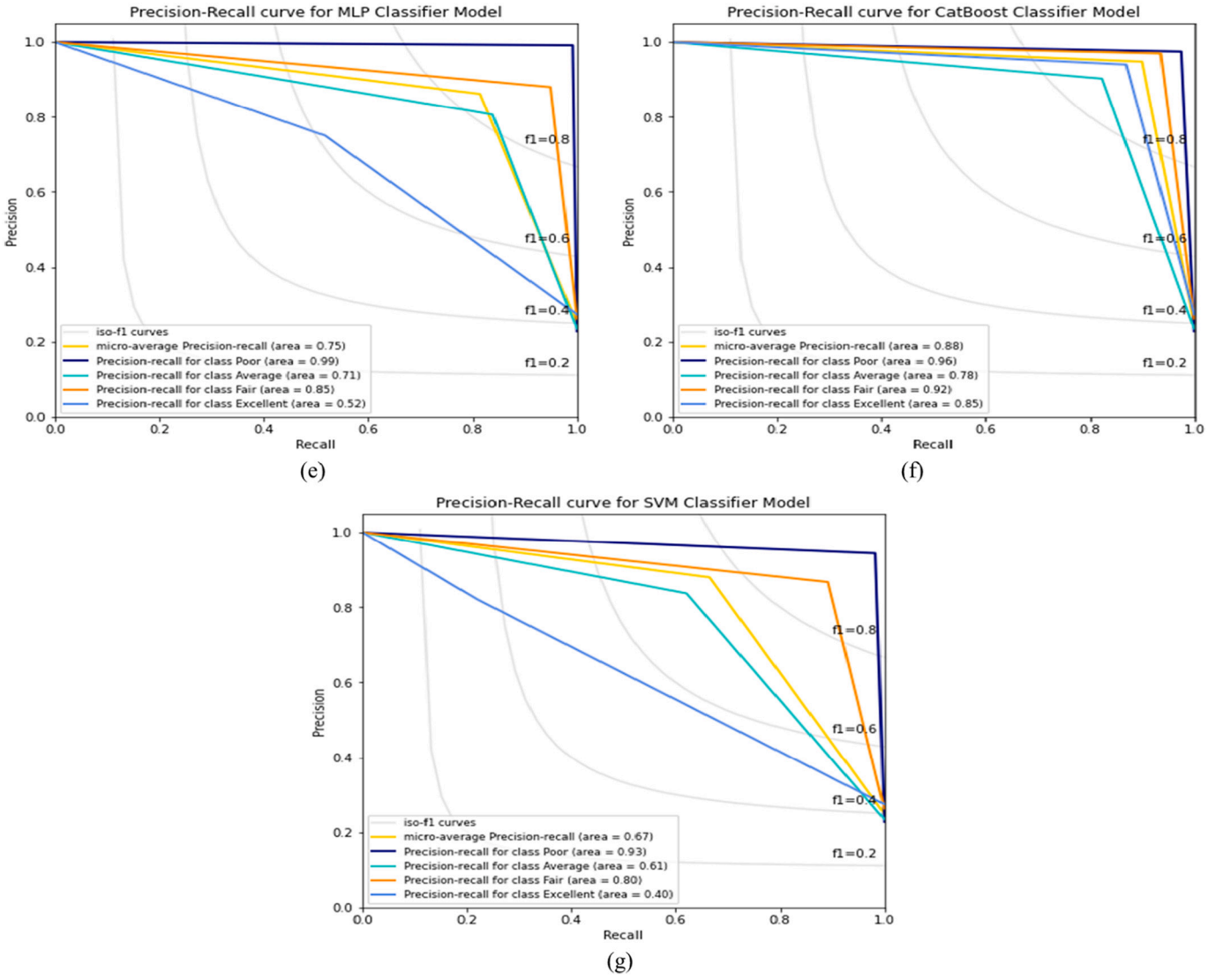


Fig. 6. (*continued*).

Table 12
Area under the curve of both ROC, and precision-recall.

	Poor (ROC | precision recall AUC)	Average (ROC | precision recall) AUC	Fair (ROC | precision recall) AUC	Excellent (ROC | precision recall) AUC
Logistic regression (AP = 0.55)	0.99 | 0.98	0.54 | 0.31	0.84 | 0.71	0.51 | 0.25
Decision tree (AP = 0.75)	0.86 | 0.89	0.96 | 0.67	0.90 | 0.80	0.87 | 0.68
Random forest (AP = 0.84)	0.96 | 0.92	0.88 | 0.75	0.92 | 0.87	0.92 | 0.85
XGB (AP = 0.75)	0.97 | 0.93	0.82 | 0.63	0.91 | 0.81	0.81 | 0.66
MLP (AP = 0.75)	0.99 | 0.99	0.98 | 0.71	0.95 | 0.85	0.73 | 0.52
CATBoost (AP = 0.88)	0.98 | 0.96	0.90 | 0.78	0.96 | 0.92	0.92 | 0.85
SVM (AP = 0.67)	0.98 | 0.93	0.79 | 0.61	0.92 | 0.80	0.60 | 0.40

$$\mathring{A}_i = \phi(x_i) = \sum_{k}^{K} f_k(x_i) \tag{11}$$

When k is the number of trees in the model and f_k is the (kth tree), we must determine the optimum collection of functions by minimizing the loss and regulation to resolve the above equation. Fig. 3 illustrates the XGBoost trees. Also, Table 8 shows the values of XGBoost used in code.

2.2.6. *Decision tree*

The decision tree (DT) algorithm has been widely utilized in several sectors since its debut as one of the most regularly used methods for data mining. A decision tree is a recursive top-down division that employs a top-down divide-and-conquer strategy. Its core algorithm is essentially Greedy. The construction of a decision tree is separated into two stages: tree building and tree pruning. The tree-building stage is the initial phase, in which a part of the training data is selected, and a decision tree is built using the breadth-first recursive method until each leaf node belongs to the same class. The second phase is the pruning stage, which utilizes the remaining data to evaluate the created decision tree and rectify any problems, prunes, and adds nodes until a proper decision tree is constructed. Pruning decreases the influence of noisy data on classification accuracy in the decision tree building method, which is a cyclical process that finally results in a decision tree. Gini index and

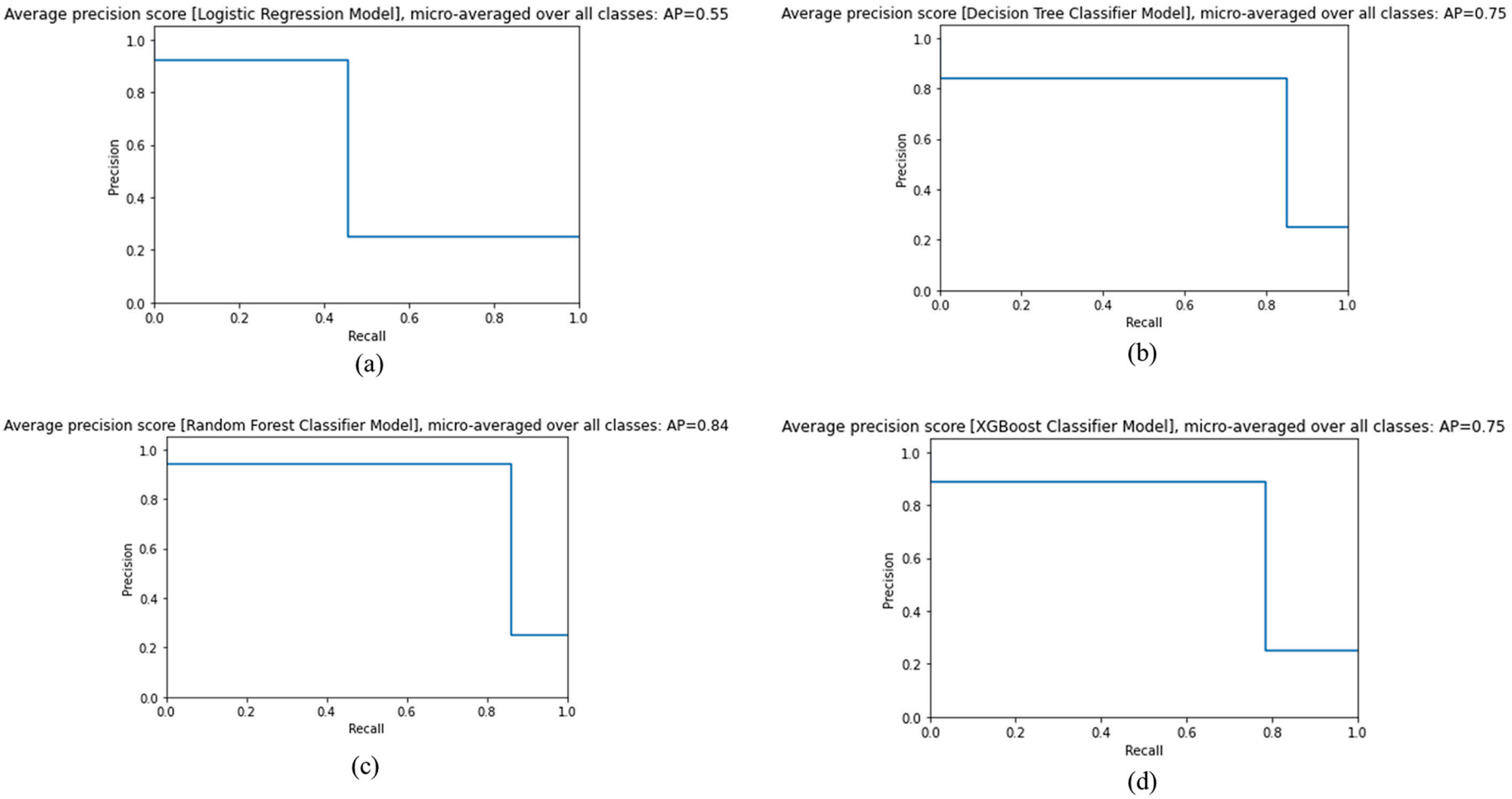
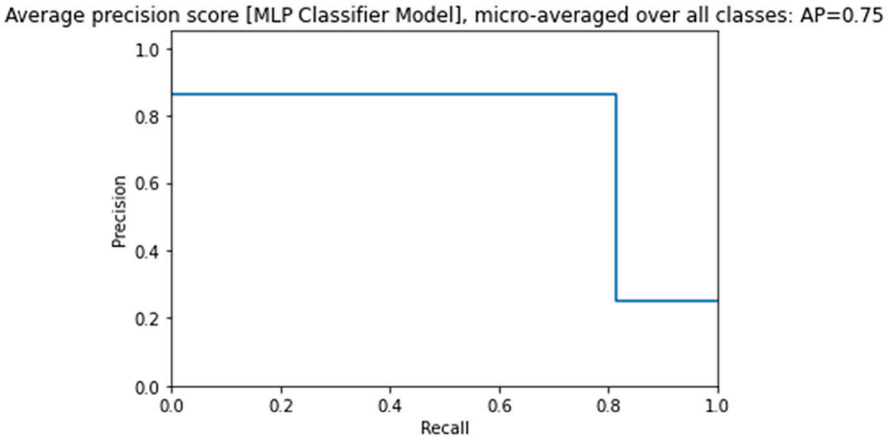
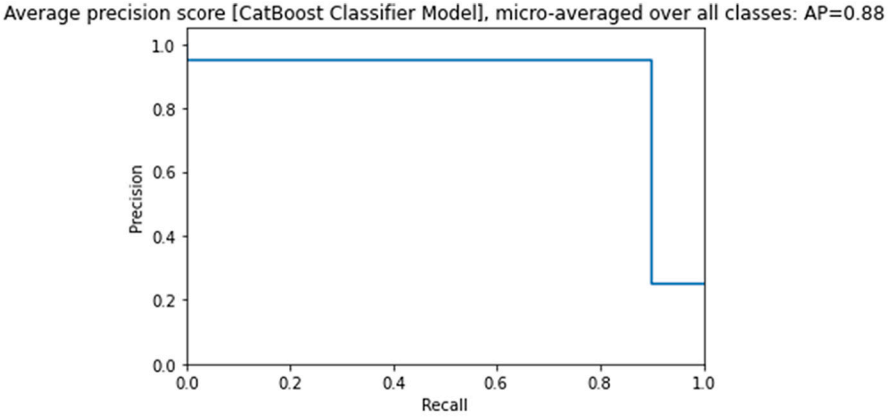


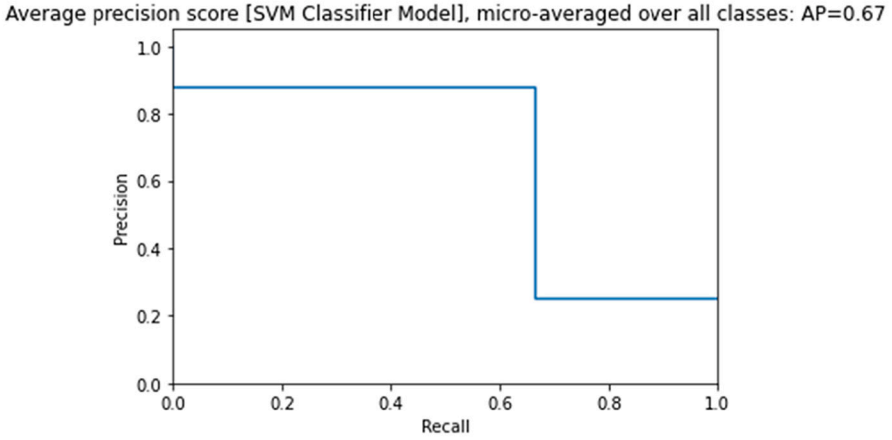
Fig. 7. Average precision score for: a) Logistic regression, b) Decision tree classifier c) Random forest classifier d) XGB classifier e) MLP classifier f) Catboost classifier g) SVM classifier.



(e)



(f)



(g)

Fig. 7. (*continued*).

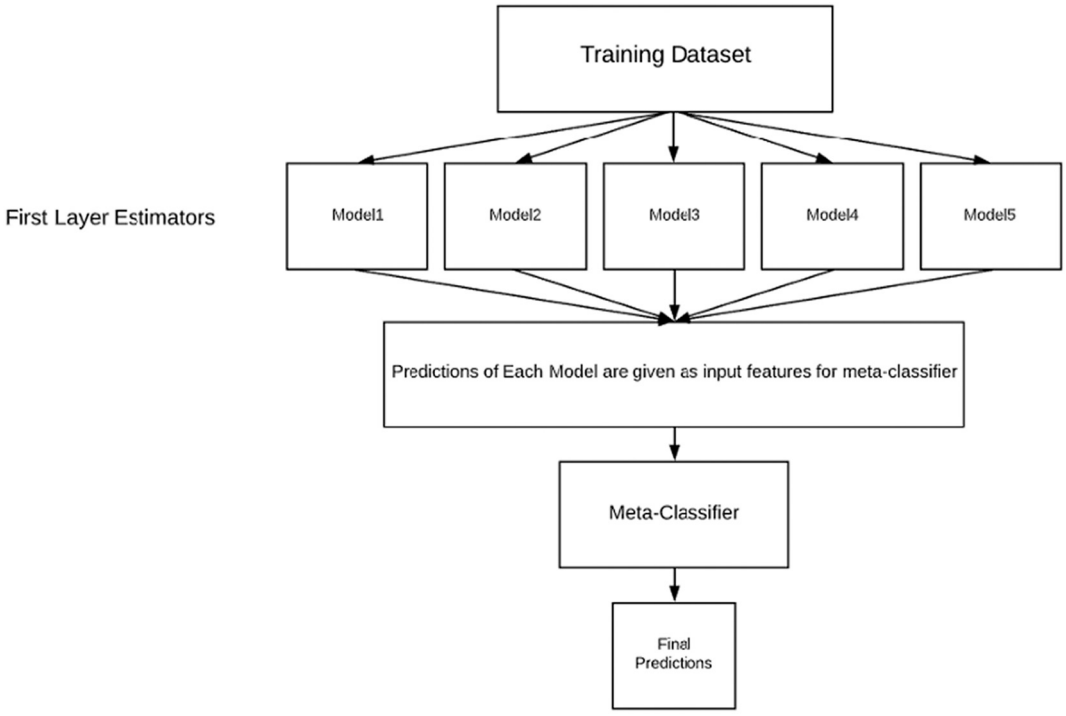


Fig. 8. The architecture of a Stacked Model.

entropy can be calculated to check the impurity of nodes [38]. Additionally, the values of LR used in code are shown in Table 9.

2.2.7. CATBoost

Boost is a gradient boosted decision tree (GBDT) and category feature-based algorithm. Under the context of the GBDT algorithm, this method is better at implementation. The critical issue is dealing with categorical characteristics efficiently and reasonably. Boost is made up of two elements: category variables and boost. This method also addresses gradient bias and prediction shift issues, enhancing the algorithm generalization ability and resilience [39]. Nonnumerical characteristics may be processed fast using Boost. When the Boost algorithm analyzes categorical features, it includes all sample data sets in the learning process. Then Boost organizes all these sample data sets at random and filters out samples from all characteristics with the same category. When numerically converting the features of each instance, the sample's goal value is computed first, followed by the weight and priority corresponding to the model [40]. Mathematically, it can be framed as follows:

$$\widehat{x}_k^i = \frac{\sum_{j=1}^{n}\left\{x_j^i = x_k^i\right\} \bullet y_i + ap}{\sum_{j=1}^{n}\left\{x_j^i = x_k^i\right\} + a} \tag{12}$$

where a is the weight coefficient, p is a prior probability distribution, y_i is the target value of the ith observation, x_k^j the original value of the kth feature for ith observation, n is the no. of observation the training set, and $\widehat{x}$ is the new value of the categorical feature. An initial value is included to reduce noise points generated by low-frequency features to avoid overfitting the model and enhance generalization capacity. The values of CATBoost used in code are shown in Table 10.

2.3. Performance measurements

Each confusion matrix describes the functioning of a classification model on a set of test data for which the actual values are well-known. The confusion matrix has been used to compute the following parameters listed in Table 11.

2.3.1. Precision

It is the proportion of correctly predicted positive observations to the total number of expected positive observations. The high accuracy is connected to the low false-positive rate. Eq. (13) demonstrates precision:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{13}$$

Precision can be improved by modifying the model parameters. However, when adjusting, it is worth noting that increasing the precision will reduce the recall, and higher recall will lead to lower accuracy.

2.3.2. Recall (sensitivity)

Relatively how many correctly predicted positives there are compared to all. Eq. (14) shows the way of calculating the recall.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{14}$$

The recall value of any ML model can be changed by altering various parameters or hyperparameters. By changing such parameters, it can either increase or decrease recall. Most positive occurrences ($TP + FN$) will be recognized if the recall is high (TP). Therefore, there will be more FP measurements and less accuracy. However, there are more FNs (positives that should have been labeled negatives) when the recall is poor, which implies that when the findings discover a positive example, it may be surer that this is indeed a positive.

2.3.3. F1-Score

The precision and recall weighted average is utilized to determine the F1 score. It is, therefore, necessary to include in the calculation of this score both false positives and false negatives not as intuitive as accuracy; F1 tends to be more useful in most cases, especially if the class distribution is uneven. Correctness is enhanced by ensuring that positive and negative results cost the same amount. If the cost of false positives and false negatives are significantly different, it is preferable to look at

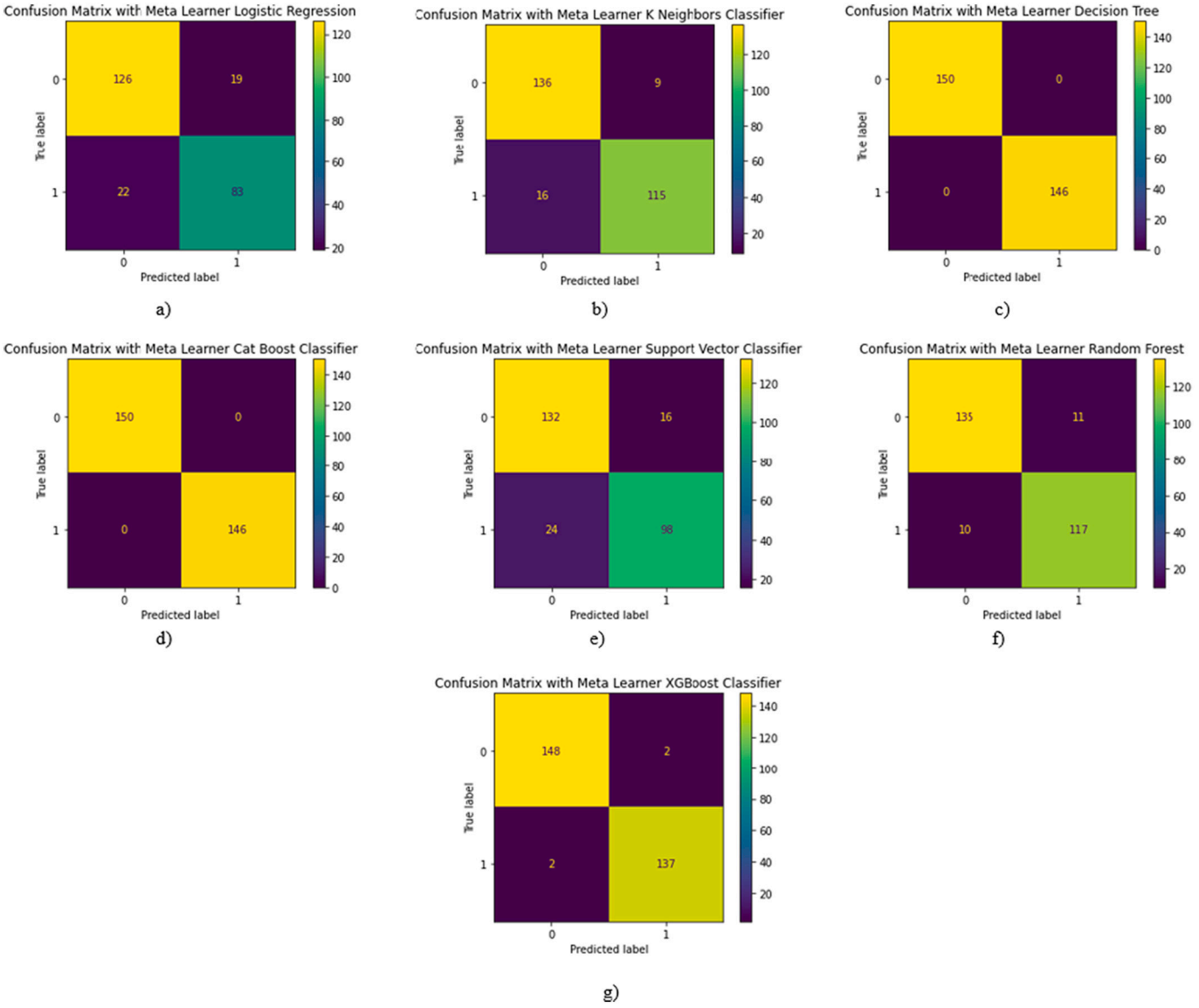


Fig. 9. Meta learner-confusion matrix: a) Logistic regression, b) KNN classifier, c) Decision tree classifier, d) CATBoost classifier, e) SVM classifier, f) Random Forest classifier, and, g) XGB classifier.

Table 13
Results of Stacked Model for various classifiers.

	Meta logistic regression	Meta SVM	Meta decision tree	Meta XGBoost	Meta MLP	Meta random forest	Meta CATBoost
Accuracy	0.6875	0.7178	1.0	0.9621	1.0	0.8106	1.0
Precision	0.6834	0.71965	1.0	0.96215	1.0	0.8108	1.0
Recall	0.6875	0.71780	1.0	0.96212	1.0	0.8106	1.0
F1 score	0.6846	0.71466	1.0	0.96208	1.0	0.8102	1.0

precision and recall.

$$F1\ Score = 2\frac{(Recall \times Precision)}{(Recall + Precision)} \tag{15}$$

2.3.4. Accuracy

The most basic and straightforward performance statistic is accuracy, just the proportion of correctly predicted observations to all observations. False-positive and false-negative rates are virtually the same in symmetrical datasets, making it a suitable quality measure. A model's performance must be evaluated in the context of other parameters.

$$Accuracy = \frac{True\ Negative + True\ Positive}{Total} \tag{16}$$

3. Results

The data has been split into 75 % for training and 25 % for testing. Fig. 4 illustrates the 4 × 4 confusion matrix for each classifier (refer to Table 2), with their color-coded values. All the performance metrics have been calculated using confusion matrices of each classifier, as mentioned in Table 11. According to the estimates, CATBoost and RF performed better than other classifiers, with a slight margin of ~1 %.

Table 14
Pros and cons of the used classifiers.

	Pros	Cons
Support Vector Machine (SVM) [54,55]	Even when there is insufficient data, it produces good outcomes.	When working with huge data sets, training takes a long time. In addition, it may be tough to perceive and comprehend due to issues created by personal circumstances and varied weights.
Random forest [56]	Fast/lower error rate/ handling of large amounts of data Extremely adaptable, with great precision. Even when a significant amount of the data is missing, it retains accuracy.	The tree predictions must be uncorrelated. It takes much time to compute, and the technique isn't very heuristic.
Multi-layer Perceptron (MLP)	It can be used to solve nonlinear issues that are difficult. Works well with big amounts of data. / Makes quick forecasts. Even with fewer data, the same accuracy ratio may be attained.	The amount to which the dependent variable influences each independent variable is unknown. Computations are time-consuming and complicated. The model's ability to operate depends on the quality of the training data. Generalization issues emerge if the model does not function effectively.
Logistic regression [57]	It has a low error Provides output probabilities It is simple to use and takes little time to train	It can't natively categorize multi-class data, but it can be customized using a variety of applications. When there are correlated attributes, it does not operate well.
XGBoost [58]	Fast to interpret/capable of handling massive datasets If the data is clean, it can avoid overfitting.	Difficult to interpret/ If the data is noisy, the model may overfit.
Decision tree	Automatic Feature Selection/ Easy Visualization	Overfitting proclivity/data sensitivity Extremely slow
CATBoost	The fitting strategy that is simple Implementation on the CPU is efficient. Model appliers may be made extremely quickly.	Too many columns needed Training is prolonged Features are less powerful in general

Followed by MLP, XGBoost, decision tree, SVM, and logistic regression.

Receiver Operating Characteristic (ROC) curves have been manifested in Fig. 5, declaring all the curve areas and average, micro/macro-average labels. The ROC curve is a familiar figure for demonstrating the trade-off between true-positive and false-positive rates for a binary classifier at various classification levels [41]. The area under the ROC curve represents the performance of a classifiers overall classification thresholds. For example, when cuddled, the upper left corner of an ideal ROC curve suggests a high actual positive rate and a low false-positive rate. ROC curves help compare various classifiers as an essential classification performance metric since they consider all potential thresholds. In addition, the ROC curve initial slope reveals how rapidly performance decreases as one moves down the data set of predictions [42].

Moreover, precision-recall curves have been implemented for all classifiers, as shown in Fig. 6. Similarly, it states different thresholds of all classifiers. It is a valuable measure of prediction success when the classes are significantly unbalanced. Precision measures result in relevancy in information retrieval, whereas recall measures how many relevant results are returned. The precision-recall curve depicts the trade-off between precision and recall [43].

A large area under the curve indicates good recall and precision, with high accuracy corresponding to a low false-positive rate and high recall corresponding to a low false-negative rate. Therefore, the area under the curves for both ROC and precision-recall curves is summarized and mentioned in Table 12. High scores for both indicate that the classifier produces accurate (high precision) results and a majority of all positive outcomes (high recall). Average accuracy is the average precision (AP) of overall potential thresholds, equivalent to the area under the precision-recall curve. It is a helpful statistic for comparing how effectively models organize predictions without taking any specific decision thresholds into account [44]; the ROC curves, which show the overlap between the classes, are constructed for each of the seven classification techniques. The ROC curves of all methods are near the upper left corner, suggesting excellent sensitivity and specificity and classifier accuracy. The ideal area under the ROC curve is constructed using the truth values of *1* and *0*, resulting in the ROC curve angle-shaped elbow. Moreover, Fig. 7 demonstrates the average precision curves for all the classifiers used. Also, Table 12 contains the score values of AP.

3.1. Model stacking

Model stacking is a technique for improving model predictions by combining the outputs of multiple models and running them through a meta-learner, which is an ML model. Stacking models work by passing the output of multiple models through a “meta-learner” (typically a linear regressor/classifier, but other models such as decision trees can also be used). All individual model's weaknesses are minimized while the meta-learner maximizes their strengths. The result is a robust model that generalizes well to new data in most cases. Fig. 8 depicts the architecture of a stacked model; however, Fig. 9 shows the meta learner-confusion matrix. Also, Table 13 demonstrates the results of the stacked model for various classifiers.

4. Discussion and future outlook

The quality of the data utilized in the learning phase impacts the outcome. Hence in this section, we will discuss the positive and negative aspects of each classifier and stacking model.

Random forests can be used for classification as well as regression. Both categorical and numerical data perform well with RF. In most cases, no scaling or change of variables is required. Random forests produce uncorrelated decision trees by performing feature selection implicitly [45]. They accomplish this by constructing each decision tree using a random collection of characteristics. This makes it an excellent model for dealing with data with many features. However, random forests are challenging to realize. They give a sense of how vital a characteristic is. However, it does not provide visibility as much into the coefficients as linear regression. In addition, for big datasets, RF can be computationally demanding [46].

On the other hand, SVM can work well when there is a distinct separating margin, in high dimensional areas, when the number of dimensions exceeds the number of samples. It is also memory efficient as it employs a subset of training points (called support vectors) in the decision function [47]. However, it does not perform well when a huge data collection is found since the necessary training time is longer. Furthermore, it does not perform well when the data set contains additional noise, such as overlapping target classes [48].

A decision tree can work with numerical and categorical features and requires minimal data preprocessing. It is fast for inference and has a non-parametric model. Its feature selection happens automatically. Nevertheless, a minute variation in data can lead to huge structure change, affecting stability. In most cases, DT takes time to process and train the model, making it expensive. Moreover, DT is insufficient for employing regression and predicting continuous values [49]. LRM is more straightforward to apply, understand, and train. It makes no assumptions about how classes are distributed in feature space. Besides, it is simple to expand to many classes (multinomial regression); and a probabilistic view of class predictions quickly [50]. Although, LRM

Table 15
Comparative study of some existing models.

Citation	Waterbody	Location	Models used	Number of parameters	Result (accuracy)
[18]	Mixed waterbodies	Pakistan	SVM, KNN, NN, and Deep NN	9	93 %
[19]	River	Malaysia	ANN	36	95.4 %
[20]	groundwater	Khuzestan Province	ANNs	16	94.0 % testing 77.0 % training
[21]	Wastewater	Iran (wastewater treatment)	multivariate linear regression (MLR) and artificial neural network (ANN)	6	74 % for MLR 79 % for ANN
[22]	Mixed waterbodies	China	RNN (Recurrent Neural Network) or LSTM (Long Short-Term Memory)	4	~94.42 %
[23]	River	India	Decision tree	6<	95.45 %
[24]	River	Ganges	Artificial Neural Network algorithms, Lavenberg Marquardt (LM), and Gradient Descent Adaptive (GDA)	2	–
[60]	River	Malaysia	FINN	25	97.7
[19]	River	Malaysia	ANNs	23	77.0
[20]	Groundwater	Iran	ANNs	16	77.0
[61]	River	Serbia	FINN	10	87.4
[62]	River	Pakistan	Polynomial Regression	4	–
[63]	Coastal waters	China	ANN, RF, CB, and SVR	12	90
[64]	River	China	EM and HC	4	–
[65]	Groundwater	India	Machine learning (Decision Makers)	–	–
[66]	Optical Water Type	Brazil	Support Vector Machines	–	94
[67]	Lake	Greece	LSTM and CNN	7	–
[68]	Rivers and lakes	India	ANFIS and FFNN	7	96.17
[69]	River	Pakistan	ANFIS (EC and TDS)	2	91 and 92
[17]	River	India	Deep Learning, ANN	3	–
[70]	River	Malaysia	WDT-ANFIS, RBF-ANN, and MLP-ANN	3	>90
[71]	Lakes	Different locations	SVM, Fuzzy	3	–
[72]	Maridalen Lake	Maridalen Lake	ANN	6	–
[73]	River		Neural network algorithm	–	98.9
[74]	River	Thailand	Data Mining	–	>90
[75]	Lakes, reservoirs, estuaries, and coastal waters	Nicaragua	CDM	–	–
[17]	Reservoirs	India	ANN, SVM, WQI, MLP, SdA, and DBN	–	–
[76]	Wastewater	Korea	ANN and SVM	5	55,100
[77]	River	Hilo Bay	learning machine and support vector regression	6	–
[78]	River water	Northern Iran	ANN	–	94.1
[44]	Rivers and lakes	China	DT, RF, and DCF	4	–
[79]	River water	Nigeria	ANN	4	99.81
[80]	Lake water	Spain	ANN	–	89
[81]	River	Iran	ANN	–	96.6
This study	Mixed water biomes	India	SVM, RF, LR, DT, CATBoost, XGBoost, and MLP.	7	100

FFNN: Feed-Forward Neural Network, ANN: Artificial Neural Network, RF: Random Forest, CB: Cubist Regression, DT: Decision Tree, SVR: Support Vector Regression, EM: Expectation-Maximization, DBN: Deep Belief Networks, DCF: Deep Cascade Forest, CDMIM: Cross-mission Data Merging and Image reconstruction with Machine learning, SVM: Support Vector Machine, SDA: Spatial Discriminant Analysis, LSTM: Long Short-Term Memory, CNN: Convolutional Neural Network, ANFIS: Adaptive-Network-Based Fuzzy Inference system, RBF: Radial Basis Function, MLP: Multi-Layer Perceptron, HC: Hierarchical Agglomerative Cluster.

should not be used if the number of observations is fewer than the number of features; otherwise, it may result in overfitting. It establishes linear boundaries. As a result, the discrete number set is linked to the dependent variable of Logistic Regression [51].

The boost methods are simple to read and understand, making prediction interpretations simple to manage. The prediction capability is effective by employing clone methods like bagging, random forest, and decision trees. Boost is a stable way to reduce over-fitting. However, because every classifier is required to repair the faults in the predecessors, boost is sensitive to outliers. As a result, the approach is overly reliant on outliers. Another problem is that scaling up the approach is tricky [52]. Each estimate is based on the accuracy of preceding predictors, making the method difficult to simplify. Boost does not require dataset conversion to any specific format like XGBoost.

Moreover, CATBoost is much faster than XGBoost. MLP can be used for complex non-linear problems and handle substantial efficient data; after training provides quick predictions. On the contrary, its computation is complex as all independent variables are affected by dependent variables. Therefore, the training phase is crucial for its fast performance [53]. Table 14 summarizes the pros and cons of each classifier used to handle the data while processing it.

Stacked generalization, also known as stacking, is an ensemble ML algorithm. It learns how to combine the predictions from two or more base ML algorithms using a meta-learning algorithm. The advantage of stacking is that it can combine the capabilities of several high-performing models to make predictions that outperform any single model in the ensemble on a classification or regression task. Using stacking models, can easily extract every bit of performance from models. Stacking models can be a quick and convenient way to achieve this in some data science problems where all kinds of performance matter. However, stacking models typically take longer to train and have much slower latencies than other models. This work can help all the researchers worldwide decide the type of classifier to be used in determining the most fitting algorithm in their study and prove the use of the stacking ensemble model. The future scope for this research cannot be limited to hardware implementation. Besides, more classifiers, neural networks, and other AI techniques can be used on the same or different datasets to enhance water monitoring [59]. Table 15 summarizes and compares the various studies related to artificial intelligence. It is seen that the proposed work performed better than other studies due to the use of the stacking ensemble model.

5. Conclusions

This research focused on classifying water quality using machine learning techniques and proposed an intelligent real-time water quality monitoring approach. Various standards and stacked ensemble models were used to provide detailed analysis to serve this purpose. The dataset included >1600 samples collected between 2005 and 2014 from various sites in India. The dataset's metadata consisted of dissolved oxygen, pH, conductivity, biochemical oxygen demand, nitrate, fecal coliform, and total coliform. The ML classifiers used in the study were support vector machine, random forest, logistic regression, CATBoost, XGBoost, decision tree, and multi-layer perceptron. Performance metrics, such as precision, sensitivity, F1 score, and accuracy, were estimated using the confusion matrix of each classifier. Moreover, precision-recall, ROC, and average precision curves were studied. In terms of precision, the order of the classifiers was as follows: CATBoost (94.51 %), random forest (94 %), followed by MLP (88.6 %), XGBoost (88.1 %), decision tree (81.6 %), SVM (80.7 %), and logistic regression (72.9 %). However, for stacking ensemble models, 100 % accuracy was delivered by multiple meta-classifiers, including decision tree, CATBoost, etc. Hence CATBoost was assessed as the best performing classifier and stacking model to enhance accuracy. With high-performance metrics, CATBoost classifier can be used in a wide range of datasets. Researchers with a good understanding of ML implementation can collaborate with experts in other fields interdisciplinary manner. The analysis can also be expanded into multiple directions, such as hardware implementation, Internet of Things (IoT), biomedical studies, and other fields.

Data availability

This dataset was obtained from the Kaggle website. Link mentioned below: https://www.kaggle.com/anbarivan/indian-water-quality-data

Declaration of competing interest

All Authors have no Conflict of Interest.

Acknowledgments

Special thanks to the Bio-Sensing Group and Research Institute of Sciences and Engineering (RISE) of the University of Sharjah, UAE for supporting this work.

References

[1] A.S. Brar, Consumer behaviour and perception for efficient water use in urban Punjab [Online]. Available: http://shodhganga.inflibnet.ac.in:8080/jspui/handle/10603/8807, 2011. (Accessed 2 September 2021).
[2] B. O'Flynn, F. Regan, A. Lawlor, J. Wallace, J. Torres, C. O'Mathuna, Experiences and recommendations in deploying a real-time, water quality monitoring system, Meas. Sci. Technol. 21 (12) (Oct. 2010), 124004, https://doi.org/10.1088/0957-0233/21/12/124004.
[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), Sep. 2015, pp. 50–54, https://doi.org/10.1109/NGCT.2015.7375081.
[4] O. Alshaltone, N. Nasir, F. Barneih, E.A. Majali, 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), Dec. 2021, pp. 174–179, https://doi.org/10.1109/DeSE54285.2021.9719474.
[5] A.Y. Sun, B.R. Scanlon, How can big data and machine learning benefit environment and water management: a survey of methods, applications, and future directions, Environ. Res. Lett. 14 (7) (Jul. 2019), 073001, https://doi.org/10.1088/1748-9326/ab1b7d.
[6] M. Bagheri, A. Akbari, S.A. Mirbagheri, Advanced control of membrane fouling in filtration systems using artificial intelligence and machine learning techniques: a critical review, Process Saf. Environ. Prot. 123 (Mar. 2019) 229–252, https://doi.org/10.1016/j.psep.2019.01.013.
[7] F. Hassanpour, S. Sharifazari, K. Ahmadaali, S. Mohammadi, Z. Sheikhalipour, Development of the FCM-SVR hybrid model for estimating the suspended sediment load, KSCE J. Civ. Eng. 23 (6) (Jun. 2019) 2514–2523, https://doi.org/10.1007/s12205-019-1693-7.
[8] M. Ehteram, S. Ghotbi, O. Kisi, A. Ahmed, G. Hayder, C. Fai, M. Krishnan, H. Afan, A. Elshafie, Investigation on the potential to integrate different artificial intelligence models with metaheuristic algorithms for improving river suspended sediment predictions, Appl. Sci. 9 (19) (Jan. 2019), 19, https://doi.org/10.3390/app9194149.
[9] N. Nasir, O.Al Bashier, A.A. Murad, M.Al Ahmad, Optical detection of dissolved solids in water samples, in: 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Nov. 2018, pp. 1–6, https://doi.org/10.1109/ICETAS.2018.8629180.
[10] N. Nasir, M.Al Ahmad, A.A. Murad, Capacitive detection and quantification of water suspended solids, in: 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Nov. 2019, pp. 1–5, https://doi.org/10.1109/ICECTA48151.2019.8959628.
[11] J. Huang, N. Liu, M. Wang, K. Yan, Application WASP model on validation of reservoir-drinking water source protection areas delineation, in: 2010 3rd International Conference on Biomedical Engineering and Informatics 7, Oct. 2010, pp. 3031–3035, https://doi.org/10.1109/BMEI.2010.5639900.
[12] Y.C. Lai, C.P. Yang, C.Y. Hsieh, C.Y. Wu, C.M. Kao, Evaluation of non-point source pollution and river water quality using a multimedia two-model system, J. Hydrol. 409 (3/4) (2011) 583–595.
[13] I.R. Warren, H.K. Bach, MIKE 21: a modelling system for estuaries, coastal waters and seas, Environ. Softw. 7 (4) (Jan. 1992) 229–240, https://doi.org/10.1016/0266-9838(92)90006-P.
[14] G. Tang, J. Li, Y. Zhu, Z. Li, F. Nerry, Two-dimensional water environment numerical simulation research based on EFDC in Mudan River, Northeast China, in: 2015 IEEE European Modelling Symposium (EMS), Oct. 2015, pp. 238–243, https://doi.org/10.1109/EMS.2015.86.
[15] E. Batur, D. Maktav, Assessment of surface water quality by using satellite images fusion based on PCA method in the Lake gala, Turkey, IEEE Trans. Geosci. Remote Sens. 57 (5) (May 2019) 2983–2989, https://doi.org/10.1109/TGRS.2018.2879024.
[16] H. Liao, W. Sun, Forecasting and evaluating water quality of chao Lake based on an improved decision tree method, Procedia Environ. Sci. 2 (Jan. 2010) 970–979, https://doi.org/10.1016/j.proenv.2010.10.109.
[17] A. Solanki, H. Agrawal, K. Khare, Predictive analysis of water quality parameters using deep learning, Int. J. Comput. Appl. 125 (9) (Sep. 2015) 29–34.
[18] U. Shafi, R. Mumtaz, H. Anwar, A.M. Qamar, H. Khurshid, Surface water pollution detection using internet of things, in: 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT (HONET-ICT), Oct. 2018, pp. 92–96, https://doi.org/10.1109/HONET.2018.8551341.
[19] N.M. Gazzaz, M.K. Yusoff, A.Z. Aris, H. Juahir, M.F. Ramli, Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors, Mar. Pollut. Bull. 64 (11) (Nov. 2012) 2409–2420, https://doi.org/10.1016/j.marpolbul.2012.08.005.
[20] M. Sakizadeh, Artificial intelligence for the prediction of water quality index in groundwater systems, Model. Earth Syst. Environ. 2 (1) (Mar. 2016) 8, https://doi.org/10.1007/s40808-015-0063-9.
[21] H.Zare Abyaneh, Evaluation of multivariate linear regression and artificial neural networks in prediction of water quality parameters, J. Environ. Health Sci. Eng. 12 (1) (Jan. 2014) 40, https://doi.org/10.1186/2052-336X-12-40.
[22] J. Liu, C. Yu, Y. Zhao, Y. Bai, M. Xie, J. Luo, Accurate prediction scheme of water quality in smart mariculture with deep Bi-S-SRU learning network, IEEE Access 8 (2020) 24784–24798, https://doi.org/10.1109/ACCESS.2020.2971253.
[23] S. Jaloree, Anil Rajput, Sanjeev Gour, Decision tree approach to build a model for water quality, Binary J. Data Min. Netw. 4 (2014) 25–28.
[24] N. Singh, Comparison of artificial neural network algorithm for water quality prediction of river Ganga, Environ. Res. J. 8 (Jan. 2014) 55–63, https://doi.org/10.3923/erj.2014.55.63, 1994-5396.
[25] A. Vasudevan, Indian water quality data. https://kaggle.com/anbarivan/indian-water-quality-data. (Accessed 19 September 2021).
[26] R.Das Kangabam, S.D. Bhoominathan, S. Kanagaraj, M. Govindaraju, Development of a water quality index (WQI) for the Loktak Lake in India, Appl Water Sci 7 (6) (Oct. 2017) 2907–2918, https://doi.org/10.1007/s13201-017-0579-4.
[27] R.Afriyie Mensah, J. Xiao, O. Das, L. Jiang, Q. Xu, M.O. Alhassan, Application of adaptive neuro-fuzzy inference system in flammability parameter prediction, Polymers (Basel) 12 (1) (Jan. 2020), E122, https://doi.org/10.3390/polym12010122.
[28] M. Kumar, A. Puri, A review of permissible limits of drinking water, Indian J. Occup. Environ. Med. 16 (1) (2012) 40–44, https://doi.org/10.4103/0019-5278.99696.
[30] X. Chen, H. Liu, F. Liu, T. Huang, R. Shen, Y. Deng, D. Chen, Two novelty learning models developed based on deep cascade forest to address the environmental imbalanced issues: a case study of drinking water quality prediction, Environ. Pollut. 291 (Dec. 2021) 118153, https://doi.org/10.1016/j.envpol.2021.118153.
[31] T.H.H. Aldhyani, M. Al-Yaari, H. Alkahtani, M. Maashi, Water quality prediction using artificial intelligence algorithms, Appl. Bionics Biomech. 2020 (Dec. 2020), e6659314, https://doi.org/10.1155/2020/6659314.
[32] Y. Xiang, L. Jiang, Water quality prediction using LS-SVM and particle swarm optimization, in: 2009 Second International Workshop on Knowledge Discovery and Data Mining, Jan. 2009, pp. 900–904, https://doi.org/10.1109/WKDD.2009.217.
[33] G. Slapničar, N. Mlakar, M. Luštrek, Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network, Sensors 19 (15) (Aug. 2019) 3420, https://doi.org/10.3390/s19153420.
[34] S.S. Bashar, A.H.M.Z. Karim, Md.S. Miah, Md.A. Al Mahmud, Z. Hasan, A machine learning approach for heart rate estimation from PPG signal using random forest

regression algorithm, in: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, Feb. 2019, pp. 1–5, https://doi.org/10.1109/ECACE.2019.8679356.
[35] A. Najah, A. El-Shafie, O.A. Karim, O. Jaafar, A.H. El-Shafie, An application of different artificial intelligences techniques for water quality prediction, IJPS 6 (22) (Oct. 2011) 5298–5308, https://doi.org/10.5897/IJPS11.1180.
[36] G.O. Krhoda, M.O. Amimo, Groundwater quality prediction using logistic regression model for Garissa County, Afr. J. Phys. Sci. 3 (Feb. 2019) 13–27.
[37] H. Lu, X. Ma, Hybrid decision tree-based machine learning models for short-term water quality prediction, Chemosphere 249 (Jun. 2020), 126169, https://doi.org/10.1016/j.chemosphere.2020.126169.
[38] C. Gakii, J. Jepkoech, A classification model for water quality analysis using decision tree [Online]. Available: http://repository.embuni.ac.ke/handle/embuni/2203, Jun. 2019. (Accessed 19 May 2022).
[39] F. Zhou, H. Pan, Z. Gao, X. Huang, G. Qian, Y. Zhu, F. Xiao, Fire prediction based on CatBoost algorithm, Math. Probl. Eng. 2021 (Jul. 2021), e1929137, https://doi.org/10.1155/2021/1929137.
[40] F.K. Abu Salem, M. Jurdi, M. Alkadri, F. Hachem, H.R. Dhaini, Feature selection approaches for predictive modelling of cadmium sources and pollution levels in water springs, Environ. Sci. Pollut. Res. (Sep.2021), https://doi.org/10.1007/s11356-021-15897-w.
[41] A. Sathyanarayana, S. Joty, L. Fernandez-Luque, F. Ofli, J. Srivastava, A. Elmagarmid, T. Arora, S. Taheri, Sleep quality prediction from wearable data using deep learning, JMIR Mhealth Uhealth 4 (4) (Nov. 2016), e6562, https://doi.org/10.2196/mhealth.6562.
[42] W. Brooks, S. Corsi, M. Fienen, R. Carvin, Predicting recreational water quality advisories: a comparison of statistical methods, Environ. Model Softw. 76 (Feb. 2016) 81–94, https://doi.org/10.1016/j.envsoft.2015.10.012.
[43] S. Chatterjee, S. Sarkar, N. Dey, S. Sen, T. Goto, N.C. Debnath, Water quality prediction: multi objective genetic algorithm coupled artificial neural network based approach, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Jul. 2017, pp. 963–968, https://doi.org/10.1109/INDIN.2017.8104902.
[44] K. Chen, H. Chen, C. Zhou, Y. Huang, X. Qi, R. Shen, F. Liu, M. Zue, X. Zou, J. Wang, Y. Zhang, D. Chen, X. Chen, Y. Deng, Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data, Water Res. 171 (Mar. 2020) 115454, https://doi.org/10.1016/j.watres.2019.115454.
[45] L. Grbcic, S. Druzeta, G. Mausa, T. Lipic, D. Lusic, M. Alvir, I. Lucin, A. Sikirica, D. Davidovic, V. Travas, D. Kalafatovic, K. Pikelji, H. Fajkovic, T. Holjevic, L. Kranjcevic, Coastal water quality prediction based on machine learning with feature interpretation and spatio-temporal analysis. https://arxiv.org/abs/2107.03230. (Accessed 19 September 2021).
[46] Z. Wang, H. Song, D. Watkins, K. Ong, P. Xue, Q. Yang, X. Shi, Cyber-physical systems for water sustainability: challenges and opportunities, IEEE Commun. Mag. 53 (5) (May 2015) 216–222, https://doi.org/10.1109/MCOM.2015.7105668.
[47] T. Xu, G. Coco, M. Neale, A predictive model of recreational water quality based on adaptive synthetic sampling algorithms and machine learning, Water Res. 177 (Jun. 2020), 115788, https://doi.org/10.1016/j.watres.2020.115788.
[48] A.S.Abobakr Yahya, A. Ahmed, F. Othman, R. Ibrahim, H. Afan, A. El-Shafie, C. Fai, M. Hossain, M. Ehteram, A. Elshafie, Water quality prediction model based support vector machine model for ungauged river catchment under dual scenarios, Water 11 (6) (Jun. 2019), 6, https://doi.org/10.3390/w11061231.
[49] N. Deepnarain, M. Nasr, S. Kumari, T. Stenstrom, P. Reddy, K. Pillay, F. Bux, Decision tree for identification and prediction of filamentous bulking at full-scale activated sludge wastewater treatment plant, Process. Saf. Environ. Prot. 126 (un. 2019) 25–34, https://doi.org/10.1016/j.psep.2019.02.023.
[50] M. Wise, W. Abrahamson, Effects of resource availability on tolerance of herbivory: a review and assessment of three opposing models, Am. Nat. 169 (Apr. 2007) 443, https://doi.org/10.2307/4137008.
[51] E. Christodoulou, J. Ma, G.S. Collins, E.W. Steyerberg, J.Y. Verbakel, B. Van Calster, A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models, J. Clin. Epidemiol. 110 (Jun. 2019) 12–22, https://doi.org/10.1016/j.jclinepi.2019.02.004.
[52] V. Uddameri, A.L.B. Silva, S. Singaraju, G. Mohammadi, E.A. Hernandez, Tree-based modeling methods to predict nitrate exceedances in the Ogallala Aquifer in Texas, Water 12 (4) (Apr. 2020), 4, https://doi.org/10.3390/w12041023.
[53] W. Chen, J. An, R. Li, G. Xie, M. Bhuiyan, K. Li, A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features, Futur. Gener. Comput. Syst. 89 (Dec. 2018) 78–88, https://doi.org/10.1016/j.future.2018.06.021.
[54] W.M. Yu, T. Du, K.B. Lim, Comparison of the support vector machine and relevant vector machine in regression and classification problems, in: ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004 vol. 2, Dec. 2004, pp. 1309–1314, https://doi.org/10.1109/ICARCV.2004.1469035. Vol. 2.
[55] L. Auria, R.A. Moro, Support Vector Machines (SVM) as a Technique for Solvency Analysis, Social Science Research Network, Rochester, NY, Aug. 2008, https://doi.org/10.2139/ssrn.1424949. SSRN Scholarly Paper 1424949.
[56] What is random forest & learn random forest using excel, New Tech Dojo, Dec. 27, 2017. https://www.newtechdojo.com/learn-random-forest-using-excel/. (Accessed 12 April 2022).
[57] J. Tu, Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, J. Clin. Epidemiol. 49 (11) (1996) 1225–1231, https://doi.org/10.1016/S0895-4356(96)00002-9.
[58] N. Kumar, The professionals point: advantages of XGBoost algorithm in machine learning, The Professionals Point, Mar. 09, 2019. http://theprofessionalspoint.blogspot.com/2019/03/advantages-of-xgboost-algorithm-in.html. (Accessed 12 April 2022).
[59] J.T. Hancock, T.M. Khoshgoftaar, CatBoost for big data: an interdisciplinary review, J. Big Data 7 (1) (Nov. 2020) 94, https://doi.org/10.1186/s40537-020-00369-8.
[60] Z. Ahmad, N.A. Rahim, A. Bahadori, J. Zhang, Improving water quality index prediction in Perak River basin Malaysia through a combination of multiple neural networks, Int. J. River Basin Manag. 15 (1) (Jan. 2017) 79–87, https://doi.org/10.1080/15715124.2016.1256297.
[61] V. Ranković, J. Radulović, I. Radojević, A. Ostojić, L. Čomić, Neural network modeling of dissolved oxygen in the Gruža reservoir, Serbia, Ecol. Model. 221 (8) (Apr. 2010) 1239–1244, https://doi.org/10.1016/j.ecolmodel.2009.12.023.
[62] U. Ahmed, R. Mumtaz, H. Anwar, A.A. Shah, R. Irfan, J. García-Nieto, Efficient water quality prediction using supervised machine learning, Water 11 (11) (Nov. 2019), 11, https://doi.org/10.3390/w11112210.
[63] S. Hafeez, M. Wong, H. Ho, M. Nazeer, J. Nichol, S. Abbas, D. Tang, K. Lee, L. Pun, Comparison of machine learning algorithms for retrieval of water quality indicators in case-II waters: a case study of Hong Kong, Remote Sens. 11 (6) (Jan. 2019), 6, https://doi.org/10.3390/rs11060617.
[64] Z. Di, M. Chang, P. Guo, Water quality evaluation of the Yangtze River in China using machine learning techniques and data monitoring on different time scales, Water 11 (2) (Feb. 2019), 2, https://doi.org/10.3390/w11020339.
[65] A. Gupta, C. Bansal, A.I. Husain, Ground water quality monitoring using wireless sensors and machine learning, in: 2018 International Conference on Automation and Computational Engineering (ICACE), Oct. 2018, pp. 121–125, https://doi.org/10.1109/ICACE.2018.8687093.
[66] E.Filisbino Freire da Silva, E. Novo, F. Lobo, C. Clemente, F. Barbosa, C. Cairo, M. Neornberg, L. Rotta, A machine learning approach for monitoring Brazilian optical water types using Sentinel-2 MSI, Remote Sens. Appl. Soc. Environ. 23 (Aug. 2021) 100577, https://doi.org/10.1016/j.rsase.2021.100577.
[67] R. Barzegar, M.T. Aalami, J. Adamowski, Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model, Stoch. Env. Res. Risk A. 34 (2) (Feb. 2020) 415–433, https://doi.org/10.1007/s00477-020-01776-2.
[68] M.Hmoud Al-Adhaileh, F.Waselallah Alsaade, Modelling and prediction of water quality by using artificial intelligence, Sustainability 13 (8) (Jan. 2021), 8, https://doi.org/10.3390/su13084259.
[69] M.I. Shah, T. Abunama, M. Javed, F. Bux, A. Aldrees, M. Tariq, A. Mosavi, Modeling surface water quality using the adaptive neuro-fuzzy inference system aided by input optimization, Sustainability 13 (8) (Jan. 2021), 8, https://doi.org/10.3390/su13084576.
[70] A.Najah Ahmed, F. Othman, H. Afan, R. Ibrahim, C. Fai, M. Hossain, M. Etheram, A. Elshafie, Machine learning methods for better water quality prediction, J. Hydrol. 578 (Nov. 2019) 124084, https://doi.org/10.1016/j.jhydrol.2019.124084.
[71] S.A. Vergina, D.S. Kayalvizhi, D.R.M. Bhavadharini, K. Devi, A real time water quality monitoring using machine learning algorithm, Clin. Med. 07 (08) (2020) 7.
[72] H. Mohammed, A. Longva, R. Seidu, Predictive analysis of microbial water quality using machine-learning algorithms, Environ. Res. Eng. Manag. 74 (Jun. 2018), https://doi.org/10.5755/j01.erem.74.1.20083.
[73] A. Kaur, M. Khurana, P. Kaur, M. Kaur, Classification and analysis of water quality using machine learning algorithms, in: Proceedings of International Conference on Communication, Circuits, and Systems, Singapore, 2021, pp. 389–398, https://doi.org/10.1007/978-981-33-4866-0_48.
[74] K. Northep, K. Srijiranon, N. Eiamkanitchat, Water quality classification using data mining techniques: a case study on Wang River in Thailand, in: 2020 Joint 9th International Conference on Informatics, Electronics Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision Pattern Recognition (icIVPR), Aug. 2020, pp. 1–8, https://doi.org/10.1109/ICIEVicIVPR48672.2020.9306655.
[75] N.-B. Chang, K. Bai, C.-F. Chen, Integrating multisensor satellite data merging and image reconstruction in support of machine learning for better water quality management, J. Environ. Manag. 201 (Oct. 2017) 227–240, https://doi.org/10.1016/j.jenvman.2017.06.045.
[76] V. Nourani, G. Elkiran, S.I. Abba, Wastewater treatment plant performance analysis using artificial intelligence – an ensemble approach, Water Sci. Technol. 78 (10) (Nov. 2018) 2064–2076, https://doi.org/10.2166/wst.2018.477.
[77] M.J. Alizadeh, M.R. Kavianpour, M. Danesh, J. Adolf, S. Shamshirband, K.-W. Chau, Effect of river flow on the quality of estuarine and coastal waters using machine learning models, Eng. Appl. Comput. Fluid Mech. 12 (1) (Jan. 2018) 810–823, https://doi.org/10.1080/19942060.2018.1528480.
[78] D.T. Bui, K. Khosravi, J. Tiefenbacher, H. Nguyen, N. Kazakis, Improving prediction of water quality indices using novel hybrid machine-learning algorithms, Sci. Total Environ. 721 (Jun. 2020), 137612, https://doi.org/10.1016/j.scitotenv.2020.137612.
[79] T.A. Folorunso, M.A. Aibinu, J.G. Kolo, S.O.E. Sadiku, A.M. Orire, Water quality index estimation model for aquaculture system using artificial neural network, J. Adv. Comput. Eng. Technol. 5 (3) (2019) 179–190.
[80] C.Doña N. Chang, V. Caselles, J. Sanchez, A. Camachi, J. Delegido, B. Vannah, Integrated satellite data fusion and mining for monitoring lake water quality status of the Albufera de Valencia in Spain, J. Environ. Manag. 151 (Mar. 2015) 416–426, https://doi.org/10.1016/j.jenvman.2014.12.003.
[81] R. Barzegar, J. Adamowski, A.A. Moghaddam, Application of wavelet-artificial intelligence hybrid models for water quality prediction: a case study in aji-Chay River, Iran, Stoch. Env. Res. Risk A. 30 (7) (Oct. 2016) 1797–1819, https://doi.org/10.1007/s00477-016-1213-y.