Hybrid Machine Learning and Optimization Algorithms for pH-Based Water Quality Classification

Xiaolin Li [1, *] and Baomeng Pang[2]
[1]School of Intelligent Manufacturing, Qingdao Huanghai University, Qingdao, Shandong, 266427, China
[2]Shandong HI-SPEED Maintenance GROUP CO., LTD, Jinan, Shandong, 250000, China
E-mail: lxl123101321@163.com
[*]Corresponding author

Keywords: water quality, PH level, machine learning, support vector classifier, extra trees classifier, optimization algorithms

Received: December 2, 2024

Water quality—defined through its physical, chemical, and biological parameters—is essential for critical applications such as drinking and irrigation. Among these parameters, pH plays a significant role by influencing metal solubility and nutrient availability, thereby impacting aquatic ecosystems. In this study, Support Vector Classifier (SVC) and Extra Trees Classifier (ETC) were employed to classify water quality based on pH values. To boost classification accuracy, the models were hybridized using two advanced metaheuristic algorithms: Transit Search Optimization Algorithm (TSOA) and Chaos Game Optimization (CGO), resulting in hybrid variants ETTS, ETCG, SVTS, and SVCG. Comprehensive experiments were conducted using standard evaluation metrics. The ETTS model achieved the best performance, with training accuracy of 0.910 and testing accuracy of 0.778, along with a precision of 0.911, recall of 0.910, and F1 score of 0.910 in training. In contrast, the base ETC model recorded training and testing accuracies of 0.881 and 0.750, respectively. Similarly, SVTS and SVCG outperformed the base SVC model, with SVTS achieving training and testing accuracies of 0.894 and 0.760, compared to SVC's 0.850 and 0.745. The proposed hybrid framework outperforms traditional SVC and ETC models and demonstrates superior classification performance compared to standard non-optimized baselines. This underscores the value of integrating advanced optimization techniques with machine learning for robust and reliable water quality assessment. The framework is a promising tool for environmental monitoring, promoting sustainable water resource management and public health protection.

Povzetek: Študija je razvila hibridne modele strojnega učenja za klasifikacijo kakovosti vode na podlagi pH-vrednosti. Kombinacija klasifikatorjev Extra Trees (ETC) in Support Vector Classifier (SVC) z metahevrističnimi algoritmi TSOA in CGO (npr. ETTS, SVTS) je izboljšala klasifikacijo. Model ETTS je dosegel najboljšo zmogljivost, kar potrjuje prednost hibridnega okvira za okoljsko spremljanje.

1 Introduction

1.1 Background

Water is as familiar a material as air, earth and concrete, Water is necessary for life for humans and other forms of life, much like the other three materials—well, maybe with exception of concrete. It is voluminous: about 3.5 % of the land area is permanently flooded, whereas two thirds of the world is under the oceans. About the hydrosphere, water is continuously evaporating from the Earth's surface into condensing in the atmosphere, reappearing as liquid. Earth's supply of water is now at an all-time high and will never be depleted [1]. Although abundant, the water resources distributed unevenly in different regions in some serious respects impede certain regions. As the population rises, industrialization increases, and even more factors such as climate change enhance problems relating to water shortages or pollution. Efficiency in water management and water quality prediction plays an important role in ensuring safety and sustainability in the use of water [2]. These are some of the issues that emanate from a lack of adequate hydrological cycles, methods of water management, and knowledge concerning the various human activities impacting catchments of water. To this end, technological and policy development remains highly critical to ensure the sustainability of the use and delivery of water, protection of public health, and economic development [3].

Water quality is basically related to its physical, chemical, and biological characteristics, making it suitable for various purposes, such as drinking, gardening, and leisure activities. During any water quality assessment, turbidity, the microbiological content, and concentrations of both organic and inorganic compounds are amongst the

more commonly measured parameters [4]. The degradation of water quality is a consequence of the current process of urbanization, agricultural runoff, and industrial wastes. Some contaminants such as heavy metals, pesticides, and viruses may result in serious human health hazards and ecosystem health. Good water grading control will require technological advancement, community participation, and regulatory mechanisms. The implementation of best practices in pollution prevention, wastewater treatment, and watershed management will ensure the sustainability of water resources through better maintenance of their quality [5].

One of the factors influencing the pH of water and hence its chemical behavior and its biological availability is the concentration of hydrogen ions in it. Basically, pH is the measure of the concentration of hydrogen ions in water. It runs on a scale from 0 to 14, with 7 to 8 being considered neutral, 0 to 7 considered acidic, and 8 to 14 considered basic. PH influences the solubility of metals and nutrients' availability, along with activity concerning aquatic organisms.

Machine learning, as a multidisciplinary subset of artificial intelligence, develops algorithms with which computers can evaluate, comprehend, and predict data [6–9]. It has powerful capabilities for identification, data analysis, and decision making and has already revamped many disciplines. The application of machine learning techniques is on the increase in environmental research to enhance our understanding and management through the modeling of environmental processes, analysis of large-scale information, and predictions of future conditions [10]. The most promising application would, therefore, be in the monitoring of water quality through management using machine learning. With the derivation of large data sets from sensors and satellite images, coupled with historical records, it will be possible for machine learning models to develop leading trends, anomalies, and predictions of water quality parameters with high accuracy [11] [12]. These capabilities enable more proactive and effective water management strategies, reducing pollution, optimizing resource allocation, and protecting public health. The integration of machine learning into the water quality monitoring system is one of the huge leaps forward in environmental science and technology [13] [14].

1.2 Research gaps and objectives

Despite the increasing application of ML in water quality prediction, significant challenges persist. Traditional approaches often struggle with the nonlinearity and complex variability of environmental data, which limits their predictive accuracy and generalizability across diverse contexts. Furthermore, while various studies have employed models like MLR, ANN, and SVM, many lack the integration of robust optimization algorithms to fine-tune model parameters and enhance performance.

Another notable gap is the underutilization of ensemble tree-based methods such as the ETC, which are known for their resilience to noise and their ability to capture intricate relationships within high-dimensional datasets. Additionally, real-time pH prediction, a critical parameter in assessing water quality, has not been extensively explored using hybrid ML-optimization techniques, especially in scenarios where both historical and real-time data are available.

To address these gaps, this study proposes a novel framework that integrates SVM, ETC, TSOA and CGO. These techniques are applied to predict and classify water pH levels using historical and sensor-based real-time datasets. The objectives of this research are:

- To develop and compare ML models capable of accurately predicting water pH levels using both historical and real-time input data;
- To optimize model performance using the Chaos Game Optimization algorithm, ensuring more reliable and efficient learning from complex datasets;
- To evaluate the classification capabilities of the Extra Trees Classifier and SVM in distinguishing water quality categories based on pH thresholds;
- To demonstrate the feasibility of a hybrid ML-optimization approach for proactive and sustainable water quality monitoring.

2 Related works

Idroes et al. [15] conducted a study to predict urban air quality in DKI Jakarta, Indonesia, using the CATBoost machine learning algorithm, which is known for handling categorical features effectively, managing missing values, and reducing the risk of overfitting. The research utilized air quality data collected from Jakarta's monitoring stations over the period of 2010 to 2021. The dataset included five key pollutants: PM_{10}, SO_2, CO, O_3, and NO_2. After a preprocessing stage that involved data cleaning and normalization, the authors split the dataset into training (80%) and testing (20%) subsets. The CATBoost model was trained and evaluated using standard performance metrics, where it achieved high accuracy (0.9781), precision (0.9722), and recall (0.9728). A feature importance analysis revealed that ozone (O_3) was the most significant contributor to air quality variation, followed by PM_{10}. Sasmita et al. [16] investigated the classification of air quality levels in Indonesia using the Plume Air Quality Index (PAQI), which incorporates pollutant concentrations such as $PM_{2.5}$, PM_{10}, NO_2, and O_3. The study focused on evaluating classification performance using Decision Tree and K-Nearest Neighbor (k-NN) algorithms, applied to secondary data collected from 33 provincial capitals between July 1 and December 31, 2022. Unlike prior studies that typically assessed model performance solely based on accuracy, this research adopted a more comprehensive evaluation approach by incorporating precision, recall, and F1-score alongside accuracy. The results demonstrated that the Decision Tree classifier outperformed k-NN, achieving performance scores of 90.67% accuracy, 90.61% precision, 90.67% recall, and 90.63% F1-score. These findings suggest that tree-based models can provide robust classification capabilities for air quality indexing, supporting more reliable monitoring and decision-making

regarding urban environmental health. Putra et al. [17] addressed the critical issue of deteriorating air quality in Indonesia's major cities, with a focus on Jakarta, where urbanization and anthropogenic activities such as vehicular emissions, industrialization, and waste accumulation have significantly impacted atmospheric conditions. Their study aimed to classify daily air quality using machine learning algorithms—specifically the C5.0 algorithm and Random Forest—based on the Air Pollution Standard Index (ISPU). These models were applied to datasets from 2017 and 2018, consisting of pollutant parameters including CO, NO_2, SO_2, PM, O_3, and NO. Their classification approach emphasized the importance of accurately identifying air quality categories to support policy-making. The models demonstrated high predictive accuracy, with C5.0 and Random Forest achieving 99.74%, 99.22%, and 99.97% accuracy on the 2017 dataset and 98.28%, 98.85%, and 97.42% on the 2018 dataset, respectively. The analysis identified O_3 (ozone) as the most influential factor in classifying air quality, with most days falling under the "Moderate" ISPU category. This work highlights the potential of decision tree-based algorithms in supporting urban air quality management through accurate pollutant classification. Saxena and Shekhawat [18] proposed a novel mathematical framework to compute a Cumulative Index (CI) for air quality classification based on the concentrations of four major pollutants: SO_2, NO_2, PM2.5, and PM10. This CI served as a compact, interpretable metric reflecting the combined impact of pollutants on air quality. Using these CI values as input features, they developed a two-class Support Vector Machine (SVM) model to classify air quality as either *good* or *harmful*. To optimize the performance of the SVM, the authors employed the Grey Wolf Optimizer (GWO) for parameter tuning, aiming to maximize classification accuracy. The methodology was tested on real datasets from three major Indian cities—Delhi, Bhopal, and Kolkata. The results indicated that the proposed classifier effectively distinguished between the two air quality categories, with high classification performance across all test locations. The study concluded that the CI-based classification framework was both computationally efficient and aligned well with actual air quality data, making it a promising tool for public health and environmental monitoring. The summary of the previous studies reported in Table 1.

Table 1: The summary of the related works.

Study	Methodology	Dataset	Metrics' results	Key Findings
Idroes et al. [15]	CATBoost machine learning for air quality prediction.	Air quality data from Jakarta monitoring stations (2010-2021). Pollutants: PM_{10}, SO_2, CO, O_3, NO_2.	Accuracy: 0.9781, Precision: 0.9722, Recall: 0.9728	Ozone (O_3) and PM_{10} most significant pollutants.
Sasmita et al. [16]	Classification using Decision Tree and k-NN algorithms.	Secondary data from 33 provincial capitals in Indonesia (2022). Pollutants: $PM_{2.5}$, PM_{10}, NO_2, O_3.	Accuracy: 90.67%, Precision: 90.61%, Recall: 90.67%, F1: 90.63%	Decision Tree outperformed k-NN for classification tasks.
Putra et al. [17]	Classification using C5.0 and Random Forest algorithms.	Air quality data (2017-2018). Pollutants: CO, NO_2, SO_2, PM, O_3, NO.	C5.0: 99.74% (2017), 98.28% (2018), RF: 99.22% (2017), 98.85% (2018)	Ozone (O_3) as the most influential factor in classifying air quality.
Saxena and Shekhawat [18]	Support Vector Machine (SVM) classification with Grey Wolf Optimizer (GWO) for parameter tuning.	Real datasets from three Indian cities (Delhi, Bhopal, Kolkata). Pollutants: SO_2, NO_2, $PM_{2.5}$, PM_{10}.	Classification performance: High accuracy for all test locations	CI-based classification framework is computationally efficient.

3 Materials and methodology

3.1 Data gathering

Water quality data were collected in a systematic manner and analyzed for different environmental parameters and their relations to pH values. The dataset used in the present study derived from [19] incorporates 1320 records in total, and each of the following input parameters has been included in the dataset: Date, Salinity, Dissolved Oxygen, Secchi Depth, Water Depth, Water Temperature, and Air Temperature. The output variable analyzed here is the pH level of the water, whether it be basic, alkaline, or acidic. Data recording over some period gathered daily data on water quality. In this case, the 'Date' variable provides for the exact day (a day in every two weeks) certain data was taken and offers a time-series track showing environmental change over time. Salinity, representing the concentration of dissolved salts in water, can directly influence pH levels by altering the ionic balance and buffering capacity of the water body. Variations in salinity may therefore contribute to shifts in pH, particularly in estuarine and coastal environments. Dissolved oxygen (DO), essential for aquatic life, can also impact pH through biological processes such as respiration and

photosynthesis, which either consume or release CO_2, thereby influencing acidity. Secchi Depth, a measure of water transparency determined by noting the depth at which a Secchi disk disappears, can serve as an indirect indicator of photosynthetic activity, which affects CO_2 levels and thus the pH. Water Depth at the sampling location affects both light availability and thermal stratification, which can influence biological activity and chemical reactions that regulate pH. Water Temperature and Air Temperature offer insight into thermal conditions that affect metabolic rates of organisms and chemical equilibria, both of which can influence pH values. The primary focus of this study was on pH levels, a key parameter in assessing water quality. In the dataset, pH values were categorized and analyzed as follows: Acidic (pH < 7) with 433 instances, Neutral (pH = 7) with 617 instances, and Basic (pH > 7) with 280 instances. Each of the variables was examined in relation to these pH categories to explore their predictive relevance.

Figure 1. consists of several parallel plots, the $x-axis$ in each plot represents the total number of samples, providing a consistent framework for comparing the distribution of each parameter. $The\ y-axis,\ varies$ according to the parameter being measured, showing the specific quantity for each sample. The red dots effectively illustrate the range and concentration of values for each parameter, offering an unambiguous graphic depiction of the data's distribution. For instance, the clustering of red dots below 0.4 meters for water depth highlights that most water samples were taken from shallow depths, with deeper samples being rare. The output pH plot illustrates the red dots form distinct horizontal bands, suggesting that pH measurements are discrete rather than continuous. This discrete distribution is crucial for classifying water quality based on pH levels.

To support the development and execution of the proposed models, a high-performance desktop workstation was utilized. This system is equipped with an Intel® Core™ i7-3770K processor clocked at 3.50 GHz and complemented by 16 GB of RAM, ensuring efficient processing and multitasking capabilities. The operating system used was Windows 11 Pro (64-bit), running on an x64-based architecture. Visual computations and graphical rendering were handled by an NVIDIA GeForce GT 640 graphics card, which contributed to a responsive and stable graphical environment. A 1 TB internal hard disk served as the primary storage medium, providing ample space for managing datasets and associated files.

All programming tasks were conducted using Python. The scikit-learn library formed the foundation for building and assessing machine learning algorithms. Data preparation and numerical analysis were facilitated by Pandas and NumPy, respectively. To aid in visual interpretation of results, Matplotlib was employed, enabling clear and informative graphical outputs throughout the analysis process.

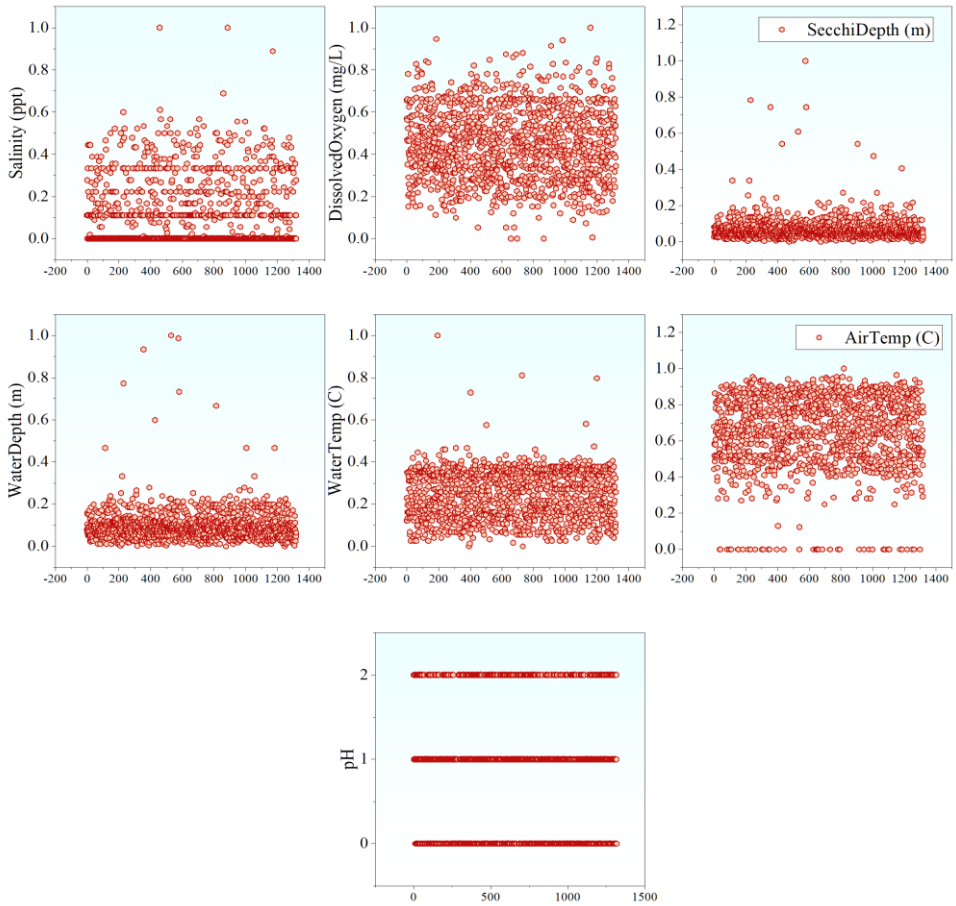


Figure 1: The parallel plot of the inputs and outputs variables

3.2 Support vector classification

Support Vector Classification (SVC) is a supervised learning algorithm rooted in the structural risk minimization principle of Support Vector Machines (SVM) [20]. It operates by mapping input features into a higher-dimensional space through non-linear kernel transformations, enabling the separation of data that is not linearly separable in the original feature space. In this transformed space, SVC constructs an optimal hyperplane that maximizes the margin — defined as the distance between the hyperplane and the closest data points from each class, known as support vectors — while simultaneously minimizing classification errors [21]. This balance between margin maximization and error minimization contributes to the model's generalization capability and robustness.

$$min_{w,b,\epsilon} \frac{\|W\|^2}{2} + C_{svc} \sum_{i=1}^{N} \epsilon_i \quad (1)$$

$$y_i(w^T.\emptyset(x_i) + b) \geq 1 - \epsilon_i \qquad i = 1,\ldots,N \quad (2)$$

$$\epsilon_i \geq 0 \qquad i = 1,\ldots,N \quad (3)$$

The function $\emptyset(x_i)$ represents a nonlinear mapping that projects each input observation x_i, defined by its explanatory variables, into a higher-dimensional feature space where linear separation of classes becomes more feasible. Within this space, w denotes the weight vector that defines the orientation of the separating hyperplane, while b is the bias term that shifts the hyperplane to achieve optimal separation. The parameter C_{svc} serves as a regularization factor that balances the trade-off between maximizing the margin and minimizing classification errors. The slack variables ϵ_i quantify the degree to which individual observations violate the margin constraints, allowing for soft-margin classification to accommodate misclassified or non-linearly separable data points.

Determining the optimal hyperplane, as formulated in Eq. (4), entails maximizing the margin between classes in the high-dimensional feature space. This objective is mathematically achieved by minimizing the Euclidean norm of the weight vector, which directly corresponds to maximizing the margin width. Simultaneously, the model incorporates a penalty for misclassified instances to ensure a balance between model complexity and classification accuracy. Ultimately, the predicted output labels indicate the class membership of each sample, based on their position relative to the decision boundary.

$$D(x_i) = W^T \varphi(x_i) + b \quad (4)$$

The computational complexity of the primal formulation is primarily dependent on the number of input features (dimensionality), whereas the dual formulation's complexity scales with the number of training samples. Therefore, in scenarios involving high-dimensional feature spaces, it is often more computationally efficient and advantageous to employ the dual form of the model, as outlined in Eqs. (5–7).

$$max_a \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} a_i a_j y_i y_j K(x_i, x_j) \quad (5)$$

$$\sum_{i=1}^{N} a_i y_i = 0 \quad (6)$$

$$0 \leq a_i \leq C_{svc} \qquad i = 1,\ldots,N \quad (7)$$

A kernel function, denoted as $K(x_i, x_j)$, computes the inner product between pairs of input samples implicitly mapped into a high-dimensional feature space, enabling nonlinear classification without explicitly performing the transformation. Common kernel types include linear, polynomial, radial basis function (RBF), and sigmoidal kernels, among others. For a kernel to be valid, it must satisfy Mercer's conditions—specifically, it must be symmetric and positive semi-definite. Extensive studies have shown that the RBF kernel, formally defined in Eq. (8), is particularly effective for classification problems due to its localized response and flexibility. Accordingly, the RBF kernel is adopted in our methodology, where the hyperparameter γ governs the inverse of the squared radius of influence of the support vectors, effectively controlling the decision boundary's smoothness and sensitivity to individual data points.

$$K(x_i, x_j) = \emptyset(x_i)^R \emptyset(x_j) = exp(-\gamma \|x_j - x_i\|) \quad (8)$$

Once the optimization process is completed and the optimal weight vector and bias term are obtained, the trained model can be used to generate predictions for unseen samples by evaluating the decision function as defined in Eq. (9).

$$SVC \quad y_i = \begin{cases} -1 \; if \; w^T \emptyset(x_i) + b \; \leq 0 \\ 1 \; if \; w^T \emptyset(x_i) + b \; > 0 \end{cases} \quad (9)$$

3.3 Extra trees classifier

The Extra trees classifier, proposed by Geurts et al. [22], represents an advanced ensemble learning technique that builds upon and extends the Random Forest framework. Unlike traditional ensemble methods that rely on bootstrapped datasets and deterministic split criteria, Extra Trees introduces two levels of randomness to enhance model diversity and generalization. First, it selects split thresholds at random rather than searching for the most optimal ones. Second, instead of using bootstrap sampling, it grows each decision or regression tree using the entire training dataset. This approach not only accelerates the training process but also reduces variance, making Extra Trees particularly effective for high-dimensional and noisy datasets.

Extra Trees operates by introducing controlled randomness into the decision tree construction process, particularly for numerical features. At each node, the algorithm selects K random features and determines split thresholds uniformly at random, rather than through traditional optimization. The minimum number of samples required to allow further splitting is defined by n_{min}, ensuring regularization. Unlike methods that rely on bootstrap resampling, Extra Trees trains each of its M trees on the entire original dataset, promoting stability and minimizing bias. For prediction, the ensemble outputs are combined using majority voting in classification tasks or averaged in regression settings. This explicit

randomization strategy—both in attribute selection and cut-point determination—significantly reduces variance and enhances generalization performance, especially in high-dimensional and noisy contexts. Although the algorithm exhibits a time complexity of $N \log N$, its computational efficiency is bolstered by the lightweight nature of the node-splitting process. The key hyperparameters—K, n_{min}, and M—govern the diversity of splits, regularization, and ensemble size, respectively. While the algorithm supports fine-tuning, default parameter configurations often yield strong performance, making Extra Trees both effective and computationally autonomous.

3.4 Chaos game optimization

The amalgamation of basic principles of chaotic games and fractals provide a mathematical model for the algorithm CGO [23]. The CGO algorithm examines several potential solutions (X) for this goal, that depicts a few suitable seeds within a sierpinski triangle, so that a group of answers that have developed by chance and selection changes is maintained by many natural evolution algorithms. According to this technique, a few chosen variables $(x_{i,j})$ reflect where these eligible seeds are located inside the triangle formed by sierpinski. with every potential solution (X_i).

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^j & \cdots & x_n^d \end{bmatrix} \quad (10)$$

According to the sierpinski triangle, where n is the number of eligible seeds and d is the seed's dimension. Based on random starting positions, these qualifying seeds are arranged in the search space.

$$x_j^j(0) = x_{i,min}^j + rand.\left(x_{i,max}^j - x_{i,min}^j\right), \quad \begin{cases} i = 1, 2, \ldots, n. \\ j = 1, 2, \ldots, d. \end{cases} \quad (11)$$

In this approach, $x_j^j(0)$ represents the initial position of qualified seeds. The values $x_{i,min}^j$ and $x_{i,max}^j$ define the lower and upper bounds for the jth decision variable of the ith candidate. A random number between 0 and 1 guides the movement direction.

Qualified seeds symbolize core concepts from chaos theory. These seeds represent candidate solutions in an optimization problem, where higher and lower fitness values indicate better and worse suitability, respectively.

To explore the search space, qualified seeds are used to construct a Sierpinski triangle—a structure made from three points: the current candidate (X_i), the group mean (MG_i), and the global best (GB). This triangle is a basis for generating new seeds using a chaos game approach.

Each triangle uses a virtual die with green and red faces to decide movement: green directs the seed toward the global best (GB), and red toward the group mean (MG_i). A random binary value (0 or 1) determines the face. This process allows seeds to move stochastically within the search space, with randomness and minimal movement controlled using factorial-based adjustments.

$$Seed_i^1 = x_i + \alpha_i \times (\beta_i \times GB - MG_i), \quad i = 1, 2, \ldots, n. \quad (12)$$

Assuming that X_i represents theith potential solution and the randomly generated factorial used to describe the limitations of seeds on movement is called α_i. To simulate the potential to roll a pair of dice β_i and γ_i stand for a random number of $0\ or\ 1$.

$$Seed_i^2 = GB + \alpha_i \times (\beta_i \times X_i - \gamma_i \times MG_i) \quad (13)$$

$$Seed_i^3 = MG_i + \alpha_i \times (\beta_i \times X_i - \gamma_i \times GB), \quad i = 1, 2, \ldots, n. \quad (14)$$

A fourth seed is produced by using an additional technique to carry out the mutation phase in the search space's position updates of the qualified seeds. This update of the seed's position is based on arbitrary modifications to the choice variables chosen at random.

$$Seed_i^4 = X_i\left(x_i^k = x_i^k + R\right), \quad k = [1, 2, \ldots, d]. \quad (15)$$

A random integer in the interval $[1, d]$ is denoted by k, and R is a uniformly distributed random number in the region $[0,1]$.

The CGO algorithm's exploration and exploitation rate can be controlled and modified by varying the movement limits of the seeds, represented by four different formulations for α_i.

$$\alpha_i = \begin{cases} Rand \\ 2 \times Rand \\ (\delta \times Rand) + 1 \\ (\varepsilon \times Rand) + (\sim \varepsilon) \end{cases} \quad (16)$$

In this case, δ and ε are random integers Rand is a random number with a uniform distribution in the interval $[0,1]$.

The process involves evaluating new seeds against existing ones to determine their eligibility for inclusion within the area used for searching. The new solution candidates' quality is evaluated, with better candidates retained and seeds with low fitness values removed. The replacement procedure is employed to simplify the mathematical model and ensure a more efficient mathematical method.

3.5 Transit search algorithm

Host star number (n_s) and the definition of signal-to-noise ratio (SN) is algorithm structure. The transit model determines SN Standard deviation of measurements made outside of transit is used to estimate noise. There is always noise in photons received from stars. The starting population for TS is equal to the product of n_s and SN [24].

- **Galaxy phase**

After identifying habitable zones, the program chooses a galactic center at random from the search space. The optimal stellar systems are found by evaluating random regions L_R. With the capacity to support life, the regions that have been identified with the best fitness are chosen, and the algorithm starts with these regions.

$$L_{R,l} = L_{Galaxy} + D - Noise \quad l = 1, \ldots, (n_s \times SN) \quad (17)$$

$$D = \begin{cases} c_1 L_{Galaxy} - L_r & if \quad z = 1 \quad (Negative\ Region) \\ c_1 L_{Galaxy} + L_r & if \quad z = 2 \quad (Positive\ Region) \end{cases} \tag{18}$$

$$Noise = (c_2)^3 L_r \tag{19}$$

L_{Galaxy} denotes where the center of the galaxy is located, and in the optimization problem, two coefficients are present ranging from $zero$ to one, denoting an accidental integer c_1 and an accidental vector c_2 representing the number of variables. To demonstrate the variation in the research area's situation, one definition of parameter D is the difference between the galaxy's center and its present condition. This region may be found either on the back of the galaxy or in the front (positive portion) of its middle area. Here, parameter zone (z) is a randomly generated number that is either one or two. The $Noise$ parameter is used to eliminate noise from received signals to improve location accuracy. To minimize computational value, the coefficient c_2 with a power of 3 is used, as noise cannot noticeably deviate from desired situations.

$$L_{s,i} = L_{R,i} + D - Noise \quad i = 1, \dots, n_s \tag{20}$$

The light spectrum (star class) that the telescope receives and the star's distance from the observer may be used to determine the luminosity of the star. It is evident that a short distance results in a higher photon count. The star's luminosity is acquired by:

$$L_i = \frac{\frac{R_i}{n_s}}{(d_i)^2} \quad i = 1, \dots, ns \quad R_i \in \{1, \dots, n_s\} \tag{23}$$

$$d_i = \sqrt{(L_s - L_T)^2} \quad i = 1, \dots, ns \tag{24}$$

Here, Star I's luminance and rank are depicted by the variables L_i and R_i. Additionally, the space between the star I and the telescope are covered by d_i. Since it is chosen at random at the beginning of the method, the location of the telescope L_T remains constant throughout the optimization.

$$L_{S,new,i} = L_{S,i} + D - Noise \quad i = 1, \dots, n_s \tag{25}$$

$$D = c_6 L_{S,i} \tag{26}$$

$$Noise = (c_7)^3 L_S \tag{27}$$

The coefficients c_6 and c_7 are a random vector from 0 to 1 and a random integer from -1 to 1. The amount of new luminosity, $L_{i,new}$ is determined by:

$$L_{i,new} = \frac{R_{i,\frac{new}{n_s}}}{\left(d_{i,new}\right)^2} \quad i = 1, \dots, ns \tag{28}$$

The new L_S and the position of the telescope may be used to compute the parameter $d_{i,new}$. It is possible to assess the possibility of transit by comparing L_i and $L_{i,new}$. If $T = 1$, the phase of the planet is utilized; if not, the phase of the neighbor e is used in this iteration.

$$\begin{aligned} &If\ L_{i,new} < L_i \quad P_T = 1 \quad (Transit) \\ &If\ L_{i,new} \geq L_i \quad P_T = 0 \quad (No\ Ttansit) \end{aligned} \tag{29}$$

This probability P_T is represented by the numbers0 (non-transit) and 1 (probability of transit). If $P_T = 1$, if the planet phase cannot be used, this iteration uses the neighbor phase.

- **Planet Phase**

Initially, at this stage, the discovered initial position of planet is identified. The quantity of light that the telescope receives decreases during a planet's transit

$$D = \begin{cases} c_4 L_{R,i} - c_3 L_r & if \quad z = 1 \quad (Negative\ Region) \\ c_4 L_{R,i} - c_3 L_r & if \quad z = 2 \quad (Positive\ Region) \end{cases} \tag{21}$$

$$Noise = (c_5)^3 L_r \tag{22}$$

The next stage involves utilizing Eq. (20) to (22) to choose a star from each of the areas that have been chosen to belong to a stellar system. L_s indicates where the stars are located. In addition to the coefficient c_5, which is a random vector between 0 and 1, the coefficients c_3 and c_4 are random values between 0 and 1.

Before beginning iterations, the suggested method executes the galaxy phase once to choose appropriate situations for the primary stages (2– 5).

- **Transit Phase**

To identify the transit, a re-measurement of the light received from the beginning is required to identify any potential decrease in the received light signals. L_S and its corresponding fitness f_S have two meanings (M_1 and M_2).

between the telescope and the star since the light comes from the star.

$$L_z = \frac{c_8 L_T + R_L L_{S,i}}{2} \quad i = 1, \dots, n_s \tag{30}$$

$$R_L = \frac{L_{S,new,i}}{L_{S,i}} \tag{31}$$

The planet's original position upon detection is demonstrated by L_z and luminance ratio is determined by R_L. Also, c_8 has a random value between 0 and 1.

$$L_{m,j} = \begin{cases} L_z + c_9 L_r & if \quad z = 1 \\ L_z - c_9 L_r & if \quad z = 2 \quad j = 1, \dots, SN \\ L_z + c_{10} L_r & if \quad z = 3 \end{cases} \quad \begin{matrix} f \\ fc \end{matrix} \tag{3 2}$$

$$L_P = \frac{\sum_{j=1}^{SN} L_{mj}}{SN} \tag{3 3}$$

To validate travel and reducing the noise's influence, one of the most crucial factors is SNThe planet's position inside its star system is specified by analyzing the quantity of signals received, which is derived from the planet's estimated position. Several SN signals are taken into account for this reason in the TS algorithm Eq. (32). The coefficient c_9 is an accidental number ranging from -1 to 1. c_{10} is a random vector with values in the range of -1 to 1. Once signals L_m have been determined, the average of SN signals are used to adjust the detected final planet position L_P. The terms Aphelion and Perihelion refer to the relative furthest and closest distances, in astronomy, between a planet (such as Earth) and the Sun or another host star. Three zones—Aphelion, Perihelion, and Neutral regions (the area between Aphelion and Perihelion areas), Eq. (32), are affected by the TS technique, which estimates the planet's orbital location using the zone parameter (z) in the planet phase.

- **Neighbor Phase**

In this phase, the present planet of the star will take its position whether the neighbor has superior circumstances compared to the current planet.

$$L_z = \frac{\left(c_{11} L_{s,new} + c_{12} L_r\right)}{2} \tag{34}$$

$$L_{n,j} = \begin{cases} L_z - c_{13}L_r & if \quad z = 1 \ \ for\ Aphelion\ region \\ L_z + c_{13}L_r & if \quad z = 2 \ \ for\ Perihelion\ region \\ L_z + c_{14}L_r & if \quad z = 3 \ \ for\ Neutral\ region \end{cases} \quad (35)$$

$$L_{N,i} = \frac{\sum_{j=1}^{SN} L_{n.j}}{SN} \quad (36)$$

Eq. (34) is used to estimate the neighbor L_z beginning position Considering its host star $L_{s,new}$ and an accidental place L_R. L_N determines the neighbor planet's ultimate position planets Eq. (35) and (36). The coefficients c_{11} and c_{12} in Eq. (41) handle a randomized integer in the range of 0 to 1. Moreover, the coefficients c_{13} and c_{14} represent a vector with a random number and a range of -1 to 1, respectively.

- **Exploitation phase**

The ideal planet for every star is identified in the earlier stages. Finding a planet by itself is meaningless. Understanding the features of the planet and the circumstances that support life is essential. This is carried out during the TS algorithm's Exploitation step. This stage expresses a revised definition of the L_P. L_P in the present phase L_E alludes to the features of the planet. Using Eq. (37), (38), the planet's ultimate properties are adjusted SN times $(j = 1, \ldots, SN)$ by adding new knowledge (K). c_{15} is an accidental number ranging from zero $to\ two$, and c_{16} is an accidental number ranging from $zero$ to one. c_{17} is an accidental vector ranging from zero to one. The knowledge index is represented by the random number c_k, which can be $1, 2, 3$, or 4. A random power between 1 and $(n_s * SN)$ is represented by P.

$$L_{E,j} = \begin{cases} c_{16}L_P + c_{15}k & if \quad c_k = 1 \quad (State\ 1) \\ c_{16}L_P - c_{15}k & if \quad c_k = 2 \quad (State\ 2) \\ L_P - c_{15}K & if \quad c_k = 3 \quad (State\ 3) \\ L_P + c_{15}K & if \quad c_k = 4 \quad (State\ 4) \end{cases} \quad (37)$$

$$K = (c_{17})^P L_r \quad (38)$$

3.6 K-Fold Cross validation

K-fold cross-validation is a widely utilized and reliable approach for evaluating and selecting models, especially in classification and regression tasks. This technique involves dividing the dataset into *k* equally sized subsets (folds). During each iteration, one-fold is reserved for validation while the remaining *k−1* folds are used for training. This process is repeated *k* times, ensuring that every subset serves once as the validation set. In this study, a 5-fold cross-validation scheme (*k = 5*) was adopted to thoroughly evaluate the proposed models and improve their generalization capability by systematically rotating the training and testing partitions. As illustrated in Fig. 2, the Support Vector Classifier (SVC) model demonstrated its peak performance during Fold 5, achieving a maximum Accuracy of 0.82. Similarly, the Extra Trees Classifier (ETC) also recorded its highest accuracy in Fold 5, with an Accuracy of 0.846, indicating consistent model performance across folds.

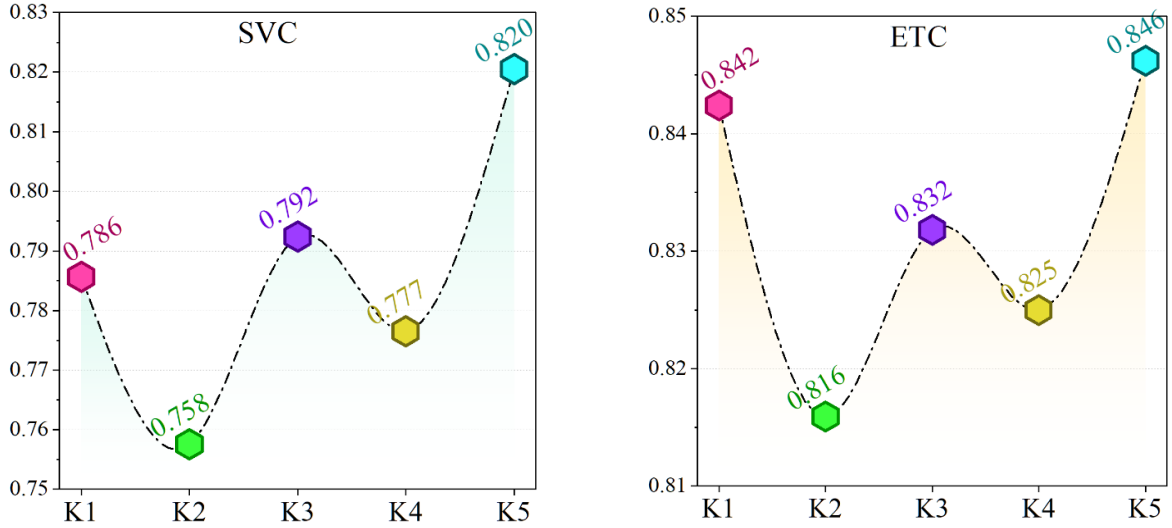


Figure 2: The results of 5-Fold Cross validation.

3.7 Evaluation metrics

The evaluation metrics of the classification models provide a quantitative measure of the performance of the models [25]. In this study, four fundamental evaluation metrics were employed to assess the performance of the classification models: Accuracy, Precision, Recall, and F1-score. These metrics provide a comprehensive understanding of model performance, especially in the context of imbalanced or complex classification problems.

- **Accuracy**

Accuracy is the ratio of correctly predicted observations to the total observations. It is a general measure of a model's effectiveness.

Accuracy serves as a baseline metric to understand the overall performance of the model. However, it may be misleading when dealing with imbalanced datasets, which is why complementary metrics are also considered.

- **Precision**

Precision is the ratio of correctly predicted positive observations to the total predicted positives. It reflects how well the model avoids false positives.

Precision is particularly valuable in scenarios were predicting a false positive may lead to unnecessary actions or costs.

- **Recall (Sensitivity)**

Recall is the ratio of correctly predicted positive observations to all actual positives. It shows how well the model detects actual positive cases. Recall is emphasized when it is more critical to identify all positive cases, even at the cost of some false positives.

- **F1-score**

The F1-score is the harmonic mean of Precision and Recall. It provides a single metric that balances both concerns, particularly useful when class distribution is uneven. The F1-score provides a consolidated metric for overall classification performance, particularly useful when neither precision nor recall alone is sufficient for model evaluation.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (39)$$

$$Precision = \frac{TP}{TP + FP} \quad (40)$$

$$Recall = \frac{TP}{TP + FN} \quad (41)$$

$$F1 - score = \frac{2.TP}{2.TP + FP + FN} \quad (42)$$

4 Result and riscussion

Prediction is actually something quite central to scientific research and practical decision-making, dealing with the estimation of the future state or event given current and historical data. Precise predictions are important in diverse fields such as meteorology to finance, for which the information furnished stands useful in planning, risk management, and policy development. Predictive models in environmental science are helpful and important to predict occurrences such as the spread of pollution, climate change, and water resource availability. The models are helpful in supporting sustainability management and conservation. Water quality prediction grounded on models such as ETC and SVC is among the most vital inputs into the planning and regulation of water quality.

Most advanced optimization techniques, such as $TSOA$ and CGO, have been employed in the enforcement of SVC and ETC for the much more improved classification of water quality according to pH. As a result, the base models ETC and SVC are involving the application of optimizers to constitute hybrid models such as $ETTS, ETCG, SVTS,$ and $SVCG$. Performance checking of the derived hybrid models is to be done for water quality prediction with respect to the pH level.

- **Hyperparameters' results**

In machine learning, hyperparameters are essential settings defined prior to training that influence model performance and learning behavior. Unlike trainable parameters, hyperparameters must be optimized to achieve the best results. In this study, random search was used to tune the hyperparameters of the proposed SVC- and ETC-based hybrid models.

As shown in Tables 2 and 3, ETC-based models were optimized using parameters such as n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_leaf_nodes. For example, ETTS used n_estimators = 143 and max_leaf_nodes = 1431, while ETCG had higher values like n_estimators = 1805 and max_leaf_nodes = 17090.

SVC-based models were tuned with C and gamma. SVTS used C = 103.098, gamma = 138.373, while SVCG had C = 679.000, gamma = 111.500. The base SVC and ETC models retained simpler, default configurations. This tuning improved accuracy and computational efficiency across all hybrid models.

Table 2: The results of Hyperparameters for ETC-based hybrid models.

Models	**Hyperparameter**				
	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_leaf_nodes
ETTS	143	143	0.001	0.000	1431
ETCG	1805	142	0.972	0.500	17090
ETC	100	None	2.000	1.000	None

Table 3: The results of Hyperparameters for SVC-based hybrid models.

Models	**Hyperparameter**	
	C	gamma
SVTS	103.098	138.373
SVCG	679.000	111.500
SVC	1.000	scale

- **Convergence curves**

Figure 3 illustrates the convergence curves of the proposed hybrid models, which combine machine learning classifiers (SVC and ETC) with metaheuristic optimization algorithms (TSOA and CGO). The figure captures the progression of classification accuracy across successive iterations, with the y-axis representing model accuracy and the x-axis denoting the number of iterations.

The convergence behavior varies notably across the hybrid configurations. The SVTS model (SVC optimized by TSOA) exhibits a steady, linear improvement in accuracy, reflecting a stable convergence pattern. In

contrast, the SVCG model (SVC optimized by CGO) demonstrates a less consistent trajectory, with noticeable fluctuations in accuracy, though an overall upward trend is still evident.

Similarly, the ETTS model (ETC optimized by TSOA) shows a smooth and consistent increase in accuracy, indicating robust convergence characteristics. The ETCG model (ETC optimized by CGO) achieves a sharper rise in accuracy, ultimately reaching a highly competitive performance level.

Among all models, ETTS achieved the highest final accuracy of 0.84, showcasing the effectiveness of the TSOA optimizer with the ETC classifier. Conversely, SVCG attained the lowest peak accuracy of approximately 0.77, suggesting less stable convergence when SVC is paired with CGO.

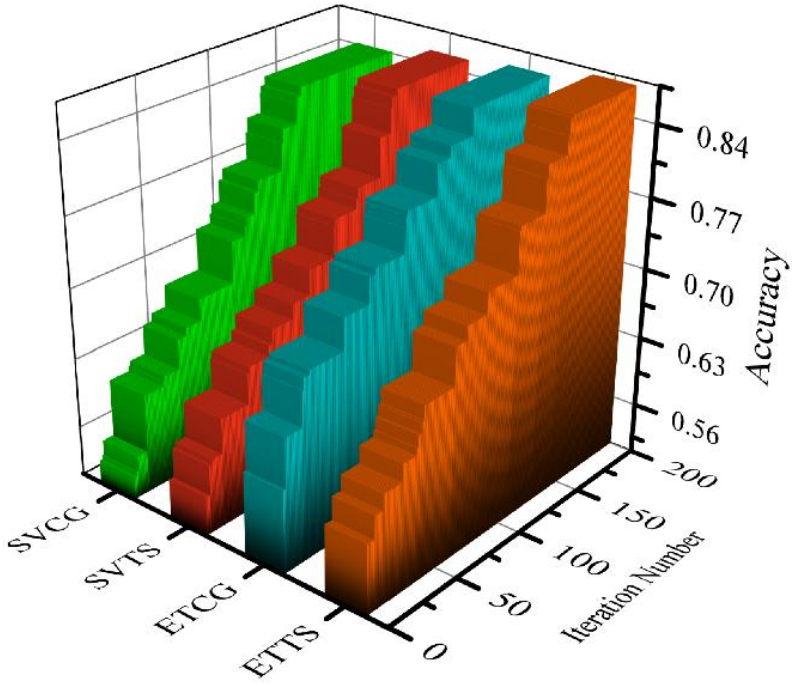


Figure 3: The convergence curve of the four presented hybrid models

Table 4 presents the performance metrics—Accuracy, Precision, Recall, and F1 Score—for both the training and testing phases of the base classifiers (ETC and SVC) and their corresponding hybrid variants (ETTS, ETCG, SVTS, and SVCG). Additionally, Figure 4 complements these results with 3D bar plots that provide a visual representation of the metric distributions for each model, highlighting comparative strengths in both learning and generalization capabilities.

Comparing the base model ETC with its hybrids, ETTS and ETCG, it is evident that both optimized variants consistently outperform the base model in both training and testing phases. For example, in the training stage, ETTS achieved the highest accuracy (0.910), followed closely by ETCG (0.897), while ETC lagged at 0.881. Similar trends are observed across Precision, Recall, and F1 Score. These performance gains continue in the testing phase, where ETTS and ETCG maintained superior generalization, with accuracies of 0.778 and 0.770, respectively, compared to ETC's 0.750.

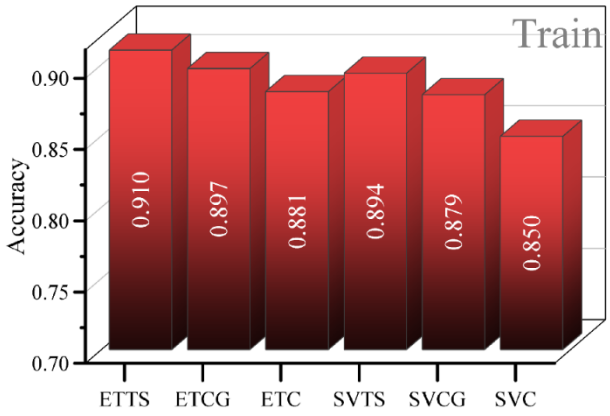
Likewise, for the SVC-based models, both SVTS and SVCG outperformed the baseline SVC during training. SVTS achieved an accuracy of 0.894, and SVCG recorded 0.879, compared to SVC's 0.850. Performance enhancements are also visible in Precision, Recall, and F1 Score. During testing, although the performance gap slightly narrows, SVTS still outpaces the base model with an accuracy of 0.760, whereas SVCG and SVC followed at 0.755 and 0.745, respectively.

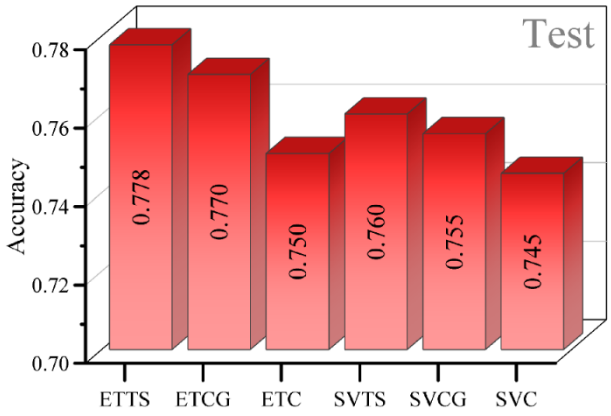
The visualized results in Figure 4 reinforces these findings. The 3D bar plots clearly illustrate the consistent superiority of hybrid models, particularly ETTS, across all evaluation metrics. The visual spacing between the bars reflects the degree of improvement, emphasizing how optimization algorithms—especially TSOA—enhance both model learning (training performance) and generalization (testing performance). The graphics also highlight that the ETTS model maintains the most balanced and highest-performing profile among all tested classifiers.

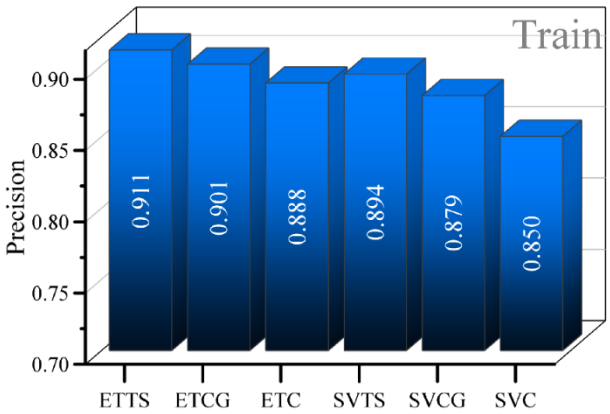
In summary, the combination of numerical evidence from Table 4 and graphical insights from Figure 4 confirms that hybrid models deliver significantly improved performance over their base classifiers. ETTS stands out as the most effective model, demonstrating the highest overall accuracy and stability across all metrics in both training and testing phases.

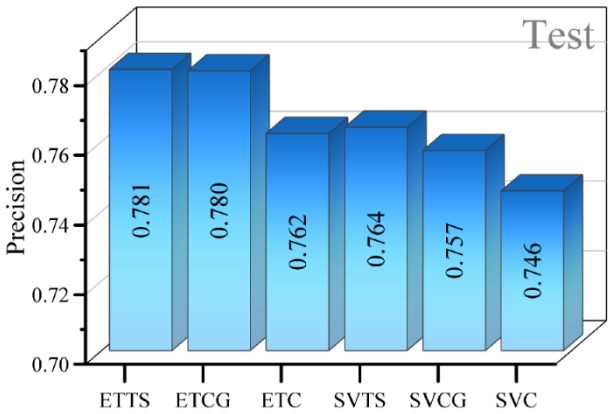
Table 4: ETC and SVC base models achieved results through the performance evaluators

Section	***Model***	**Metrics**			
		Accuracy	Precision	Recall	F1 Score
Training	ETTS	0.910	0.911	0.910	0.910
	ETCG	0.897	0.901	0.897	0.897
	ETC	0.881	0.888	0.881	0.880
	SVTS	0.894	0.894	0.894	0.894
	SVCG	0.879	0.879	0.879	0.879
	SVC	0.850	0.850	0.850	0.849
Testing	ETTS	0.778	0.781	0.778	0.778
	ETCG	0.770	0.780	0.770	0.769
	ETC	0.750	0.762	0.750	0.749
	SVTS	0.760	0.764	0.760	0.760
	SVCG	0.755	0.757	0.755	0.755
	SVC	0.745	0.746	0.745	0.745









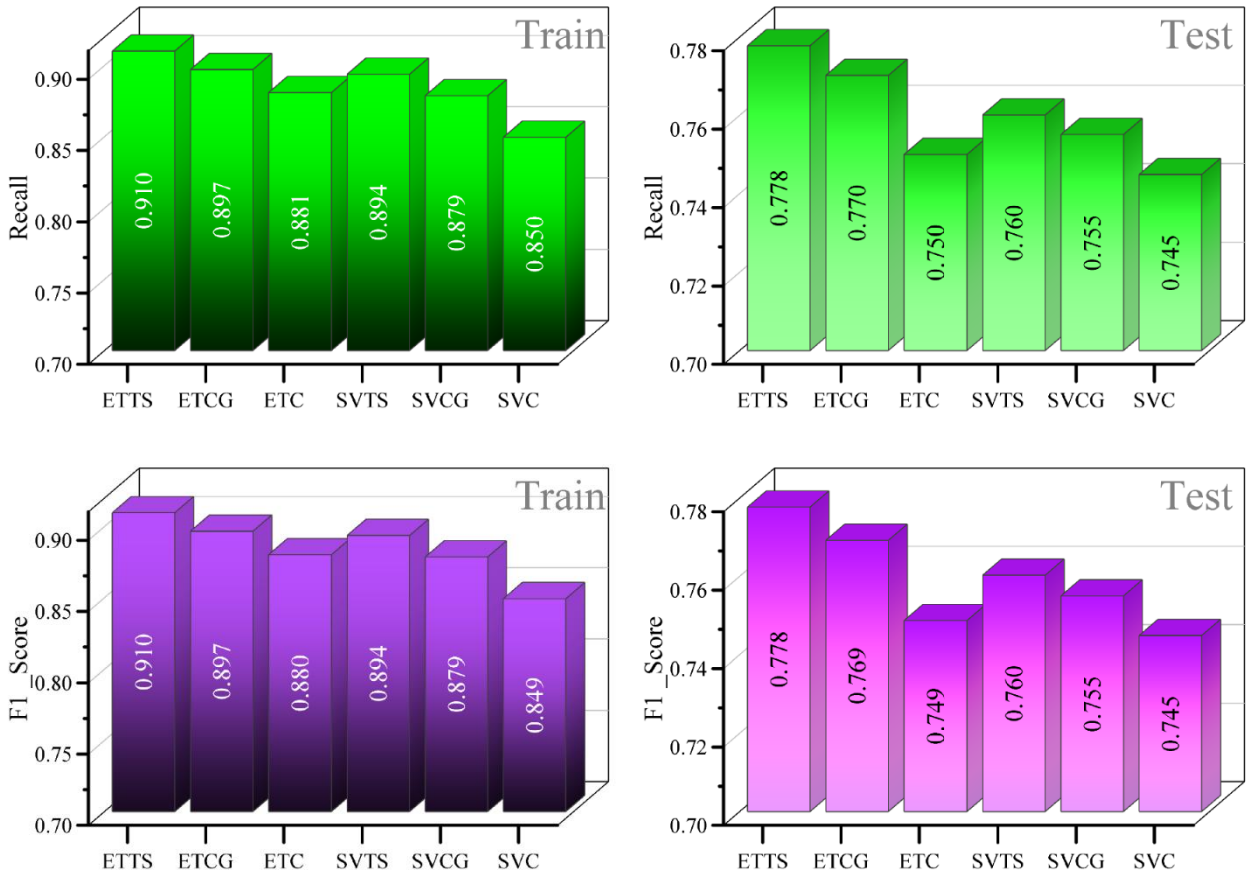


Figure 4: 3D bar plot for the performance of the models in train and test phases.

Table 4 outlines the performance metrics of base models and hybrid models. In a similar manner, Table 5 present the models' precision, Recall, and $F1\ score$ but in a more detailed breakdown of machine learning models that applied to water quality classification based on pH levels and categorized into Acidic, Basic, and Neutral conditions. The performance comparison of ETC with $ETTS$ reveals significant improvements across all pH conditions. For the Acidic condition, ETC displays an $F1\ score$ of 0.842, recall of 0.804, and precision of 0.883 , whereas $ETTS$ improves these metrics to 0.874 in precision, 0.868 in recall, and 0.871 in $F1\ score$. In the Basic condition, with a precision of 0.919, recall of 0.732, and $F1\ score$ of 0.815, ETC trails behind $ETTS$, which performs better with a precision of 0.890, recall of 0.807, and $F1\ score$ of 0.846. ETC reports an $F1\ score$ of 0.852, recall of 0.919, and precision of 0.794 for the Neutral condition. Whereas $ETTS$ achieves higher scores with 0.860 in precision, 0.901 in recall, and 0.880 in $F1\ score$. These numbers highlight the enhanced performance of $ETTS$, particularly in recall and $F1\ scores$, demonstrating the effectiveness of optimization. Both ETC and SVC show substantial improvements in precision, recall, and $F1\ scores$ when optimized with $TSOA$ and CGO, respectively. For instance, in the acidic condition, SVC achieves a precision of 0.800 while $SVTS$ outperforms SVC by improvement in precision to 0.865. The optimized models demonstrate superior capability in accurately classifying water quality, with $ETTS$ and $ETCG$ performing notably well in various metrics. Among all the models evaluated, the $ETTS$ model emerges as the best performer, achieving the highest overall accuracy in $pH-based$ water quality classification.

Table 5: Model performance in the three different conditions

Model	Condition	Metric			P-value
		precision	recall	f1-Score	
ETTS	Acidic	0.874	0.868	0.871	0.032
	Basic (alkaline)	0.890	0.807	0.846	0.027
	Neutral	0.860	0.901	0.880	0.018
ETCG	Acidic	0.887	0.834	0.860	0.04
	Basic (alkaline)	0.922	0.764	0.836	0.035
	Neutral	0.821	0.921	0.868	0.022
ETC	Acidic	0.883	0.804	0.842	0.045
	Basic (alkaline)	0.919	0.732	0.815	0.039

	Neutral	0.794	0.919	0.852	0.025
SVTS	Acidic	0.865	0.841	0.853	0.048
	Basic (alkaline)	0.841	0.811	0.826	0.041
	Neutral	0.852	0.883	0.867	0.029
SVCG	Acidic	0.840	0.825	0.832	0.052
	Basic (alkaline)	0.827	0.804	0.815	0.047
	Neutral	0.849	0.872	0.860	0.031
SVC	Acidic	0.800	0.801	0.801	0.059
	Basic (alkaline)	0.814	0.764	0.788	0.053
	Neutral	0.833	0.855	0.844	0.010

Figure 5 depicts a line plot illustrating the numerical differences in how well different machine learning models perform when used to classify water quality based on pH. This figure's main purpose is to compare various models' efficaciousness visually. Particularly focusing on the performance improvements achieved by incorporating sophisticated optimization algorithms. ETC and its hybrid version, $ETTS$, show distinct differences. ETC correctly predicts 558, 348, and 205 samples in neutral, acidic, and alkaline groups. While $ETTS$ improves upon this with a predicted value of 547, 376, and 226 samples in neutral, acidic, and alkaline, indicating an enhancement in accuracy. This improvement is quantified as a percentage difference in the accuracy of the models, with $ETTS$, in general, showing lower percentage differences compared to ETC, highlighting its enhanced predictive capability.

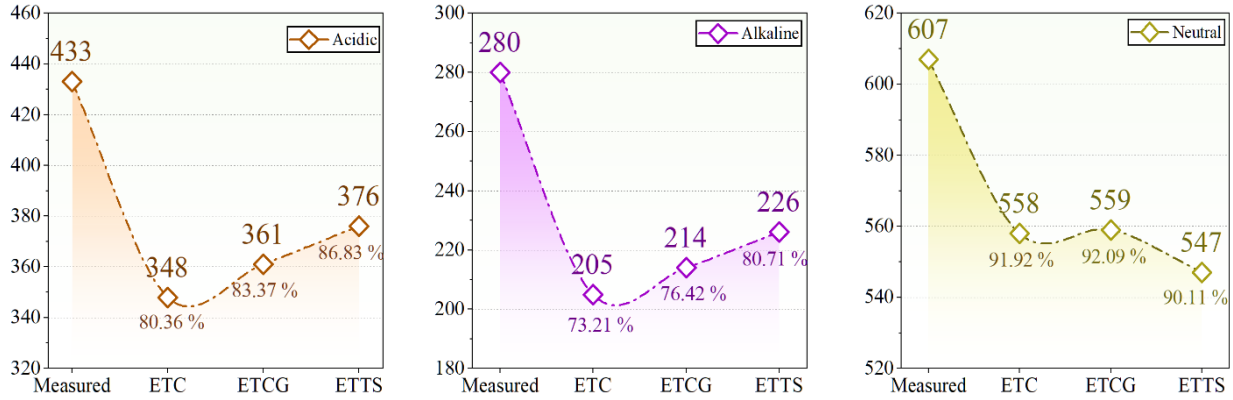


Figure 5: Line plot representing the number of correct predictions by ETC-based models

A comprehensive evaluation of each model's accuracy can be done thanks to the confusion matrix, which is depicted in Figure 6 and compares actual and predicted classifications. An illustration of the confusion matrix created by different machine-learning models for determining the pH-level-*based* classification of water quality is shown in Figure 6. Each model's accuracy can be thoroughly evaluated thanks to the confusion matrix, which displays actual versus predicted classifications. ETC predicts acidic samples with 348 correct, three misclassified as alkaline, and 82 as neutral. For alkaline samples, it predicts 205 samples correctly, with 12 samples misclassified as acidic and 63 samples as neutral. Neutral samples are predicted, with 558 samples correctly, 34 as acidic, and 15 as alkaline. When optimized using the Transit Search Optimization Algorithm, the hybrid model ($ETTS$) shows improved performance. $ETTS$ predicts acidic samples with 376 correct, seven misclassified as alkaline, and 50 as neutral. For alkaline samples, $ETTS$ predicts 226 correctly, with 15 misclassified as acidic and 39 as neutral. Neutral samples are predicted with 547 correctly, 39 as acidic, and 21 as alkaline. Comparatively, the $ETTS$ model outperforms its base model ETC, especially in predicting neutral samples with significantly higher accuracy. In acidic classification, $ETTS$ shows slight improvement with fewer misclassifications. For alkaline predictions, both models show comparable performance, though $ETTS$ has a marginally better accuracy. Among all models, the best performance is observed in the $ETTS$ model, indicating its superior capability in accurate $pH-based$ water quality classification.

	Acidic	Alkaline	Neutral
Acidic	376	7	50
Aalkaline	15	226	39
Neutral	39	21	547

ETTS

	Acidic	Alkaline	Neutral
Acidic	361	4	68
Aalkaline	12	214	54
Neutral	34	14	559

ETCG

	Acidic	Alkaline	Neutral
Acidic	348	3	82
Aalkaline	12	205	63
Neutral	34	15	558

ETC

Figure 6: Confusion matrix for the accuracy of each model.

To evaluate the classification performance of the models in predicting pH-based water quality, the Receiver Operating Characteristic (ROC) curves in Figure 7 are analyzed. These curves illustrate the trade-off between the true positive rate and the false positive rate at various threshold settings, providing a visual assessment of each model's diagnostic ability.

The micro-average ROC curve (green dashed line) aggregates the contributions of all classes, treating each prediction equally. It reflects the classifier's overall ability across all samples. The curve's steep initial rise indicates strong overall performance, with high sensitivity achieved at low false positive rates.

The macro-average ROC curve (red dashed line) calculates the average performance across classes by assigning equal weight to each one, regardless of class imbalance. It provides a balanced view of performance and shows a smoother increase in true positive rate compared to the micro-average.

Performance across specific pH categories is also shown:

- The acidic class (brown line) demonstrates moderate sensitivity at the outset, improving with higher false positive rates.
- The basic (alkaline) class (cyan line) exhibits the most favorable curve, with a sharp ascent indicating excellent classification performance at low false positive rates.
- The neutral class (purple line) shows a more gradual increase, reflecting a balanced but less pronounced trade-off between true and false positives.

Overall, the cyan curve representing basic pH conditions shows the highest classification accuracy, while the green micro-average curve confirms the robustness of the models in handling all classes collectively.

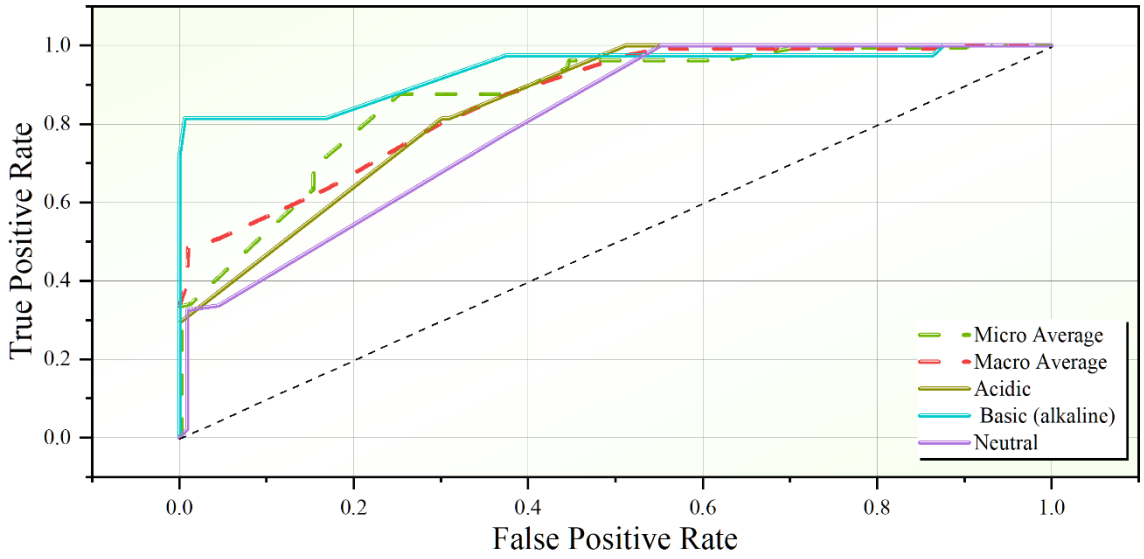


Figure 7: The ROC curves for the performance of the most efficient hybrid models

- **Wilcoxon test**

Figure 8 presents a radar plot of the Wilcoxon test statistics for all single and hybrid models: SVC, SVTS, SVCG, ETC, ETTS, and ETCG. The plotted values reflect the Wilcoxon test statistic for each model when compared pairwise, quantifying relative performance in terms of statistical ranking.

From the figure:

- SVC records the highest Wilcoxon statistic (13,521), indicating that its performance significantly differs—statistically outperforming or underperforming—relative to others.
- ETTS also scores high (12,648.5), suggesting a strong and consistent performance validated by statistical evidence.
- In contrast, SVTS and SVCG have lower statistics (9313 and 10,945.5, respectively), pointing to less statistical dominance or more variability across comparisons.
- ETCG and ETC show intermediate values (7725 and 10,063.5), reflecting moderate performance consistency.

The shaded blue region visually represents the distribution and spreads of the Wilcoxon test statistics across all models. A wider area suggests higher variability in model ranks, while more compact regions suggest more stability.

Overall, the Wilcoxon analysis complements accuracy-based evaluation by statistically confirming the comparative significance of the observed model performance differences.

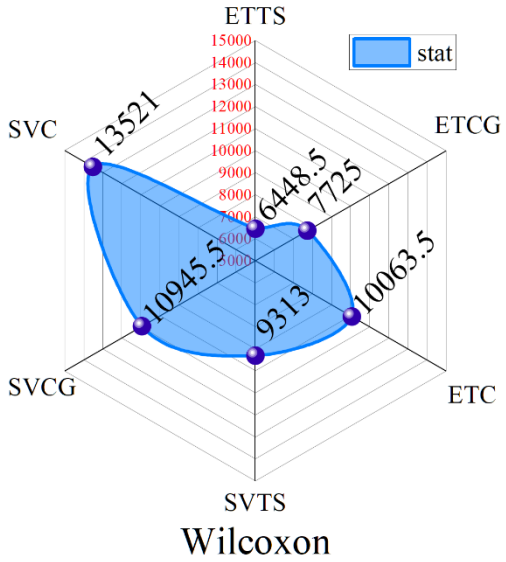


Figure 8: The results of Wilcoxon test for models' performance.

5 Discussion

5.1 Limitations of the study

While the proposed hybrid models (ETTS, ETCG, SVTS, and SVCG) demonstrated superior classification performance over their baseline counterparts, the study presents several limitations that warrant attention. First, the dataset used for model training and evaluation comprises only 1,320 daily records, which may limit the generalizability of the models across diverse geographical regions or seasonal variations. A larger and more heterogeneous dataset could improve robustness and reduce the risk of overfitting. Secondly, the models focus solely on pH as the output classification parameter, potentially neglecting the complex interactions of other water quality indicators (e.g., turbidity, nitrate levels) that may jointly influence classification outcomes.

5.2 Potential future studies

Building upon the promising results of this study, future research can explore several enhancements. One key direction is the expansion of the dataset, both temporally and spatially, to include diverse water bodies, seasonal dynamics, and additional environmental indicators. This would allow for the training of more generalizable models applicable to broader real-world conditions. Additionally, the integration of deep learning architectures—such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs)—can be investigated for their potential to capture temporal or spatial correlations in water quality trends. Furthermore, an ensemble framework combining multiple hybrid models could be tested using voting or stacking strategies to further improve classification performance.

5.3 Practical implications of the study

The findings of this study highlight the practical viability of hybrid machine learning and optimization frameworks in environmental monitoring applications. By accurately classifying water quality based on pH levels, the proposed models can assist water resource managers, environmental agencies, and public health officials in making informed decisions regarding water treatment and ecosystem preservation. The enhanced predictive accuracy of the hybrid models ensures timely identification of acidic or alkaline deviations, which are critical for preventing metal toxicity, preserving aquatic biodiversity, and maintaining water usability for irrigation and drinking purposes. Moreover, the lightweight nature of the models (especially ETC and SVC) makes them suitable for deployment in embedded or real-time monitoring systems, offering

scalable solutions for smart water quality surveillance in both urban and rural settings.

5.4 Comparison between the results of present study and previous works

Table 6 presents a comparative analysis between the proposed hybrid model (ETC+TSOA) from the present study and several existing state-of-the-art methods in the domain of water quality classification. The comparison is based on classification accuracy, which is a key performance metric. Among the referenced studies, Putra et al. [17] achieved the highest accuracy (0.9828) using a Random Forest Regressor (RFR), followed closely by Idroes et al. [15] with a CATBoost model (0.9781). Sasmita et al. [16] employed a K-Nearest Neighbors (KNN) classifier and reported an accuracy of 0.9067. In contrast, the present study's ETC+TSOA model attained an accuracy of 0.91, outperforming the KNN-based model and demonstrating competitive results relative to more complex ensemble methods.

While the accuracy of the ETC+TSOA model is slightly lower than that of RFR and CATBoost, it is important to note that the proposed model leverages advanced metaheuristic optimization to enhance model performance while maintaining a balance between interpretability, computational efficiency, and generalization capability. This underscores the value of hybrid machine learning and optimization approaches, especially in resource-constrained or real-time environmental monitoring contexts.

Table 6: The Comparison between the results of present study and previous works.

Article	Reference	Model	Metrics
			Accuracy
Idroes et al.	[15]	CATBoost	0.9781
Sasmita et al.	[16]	KNN	0.9067
Putra et al.	[17]	RFR	0.9828
Present study	**-**	**ETC+TSOA**	**0.91**

6 Conclusion

Water quality is a very important aspect in which environmental health and safety can be ensured. For understanding aquatic ecosystems for the purpose of monitoring and management, proper classification of water quality is required, mainly based on their pH levels. This research article applied various methods of artificial intelligence and optimization algorithms for the categorization of the quality of water based on pH levels, hence providing a robust framework for environmental monitoring. In this research, the dataset used contains 1320 records in total; each record has information on the following input parameters: Date, Salinity, Dissolved Oxygen, secchi Depth, Water Depth, Water Temperature, and Air Temperature. The output parameter in this analysis is pH, or the level of acidity, alkalinity, and neutrality indicative of water. These are daily records; hence, they provide a holistic view of how the respective environmental matters are changing from day to day.

In the presented study, SVC and ETC were used for water quality prediction by considering pH as one of the main influential parameters. In the present study, a more advanced class of optimizers in the form of the Transit Search Optimization Algorithm and Chaos Game Optimization were coupled with the svcand ETC to improve their corresponding predictive accuracies. The obtained results reflected that the hybrid models $ETTS$, $ETCG$, $SVTS$, and $SVCG$ outperformed their base model with a significant difference in performance.

Comparing $ETTS$, when all models are taken into consideration against the ETC base model, it improves Accuracy by 3.73%, with increased Precision by 2.49%, boosted Recall by 3.73%, and increased $F1\,Score$ by 3.87%. On the other hand, $ETCG$ outperforms ETC with improved Precision by 2.36%, increased Accuracy and Recall by 2.67%, and a better $F1\,Score$ by 2.67% also. For SVC models, $SVTS$ increased Accuracy and Recall by 2.01%, increased Precision by 2.41%, and also increased the $F1\,Score$ by 2.01% from the base SVC. Similarly, $SVCG$ also outperformed SVC, with increases of 1.34% in Accuracy and Recall, and it boosted Precision by 1.47%. $ETTS$ turned out to be the best improvement among all, with the highest scores on all metrics.

High capability of hybrid models to provide more reliable and accurate pH-based water quality prediction underlines the potential for such advanced techniques in environmental monitoring and management. These results demonstrate how combining machine learning with advanced optimization algorithms yields significantly higher predictive accuracy and reliability for pH-based water quality classification. The usefulness of hybrid models in these applications, due to their increased accuracy, makes them very handy tools in the prediction of water quality, therefore helping in water body management and conservation.

References

[1] Boyd, C.E (2019). *Water quality: an introduction*. Springer Nature.

[2] Mekonnen, M.M. and A.Y. Hoekstra (2016). Four billion people facing severe water scarcity. *Science Advances*, Science, 2(2), p. e1500323. https://doi.org/10.1126/sciadv.1500323.

[3] Vorosmarty, C., P. Green, J. Salisbury and R. Lammers (2000). Global Water Resources: Vulnerability from Climate Change and Population Growth. *Science*, Science, 289, p. 284. https://doi.org/10.1126/science.289.5477.284.

[4] Chapman, D (1992). *Water Quality Assessments - A Guide to Use of Biota, Sediments and Water in Environmental Monitoring - Second Edition*. Taylor & Francis, https://doi.org/10.1201/9781003062103.

[5] Schwarzenbach, R., B. Escher, K. Fenner, T. Hofstetter, C. Johnson, U. Gunten and B. Wehrli (2006). The Challenge of Micropollutants in Aquatic Systems. *Science (New York, N.Y.)*, Science, 313, pp. 1072–1077. https://doi.org/10.1126/science.1127291.

[6] Yang, X (2025). Economic Cost Prediction Model for Building Construction Based on CNN-DAE Algorithm. *Informatica*, Slovenian Society Informatika, 49(5). https://doi.org/10.31449/inf.v49i5.7029.

[7] Dash, C.S.K., S.C. Nayak, A.K. Behera and S. Dehuri (2023). A Neuro-Fuzzy Predictor Trained by an Elitism Artificial Electric Field Algorithm for Estimation of Compressive Strength of Concrete Structures. *Informatica*, Slovenian Society Informatika, 47(5). https://doi.org/10.31449/inf.v47i5.3951.

[8] Benkaddour, M.K (2021). CNN based features extraction for age estimation and gender classification. *Informatica*, Slovenian Society Informatika, 45(5). https://doi.org/10.31449/inf.v45i5.3262.

[9] Maktum, T., N. Pulgam, V. Chandgadkar, P. Pathak and A. Solanki (2025). A Machine Learning Based Framework for Bankruptcy Prediction in Corporate Finances Using Explainable AI Techniques. *Informatica*, Slovenian Society Informatika, 49(15). https://doi.org/10.31449/inf.v49i15.6745.

[10] Mitchell 1951-, T.M (1997). *Machine Learning*. McGraw-Hill.

[11] Al-Adhaileh, M. and F. Alsaade (2021). Modelling and Prediction of Water Quality by Using Artificial Intelligence. *Sustainability*, MDPI, 13, p. 4259. https://doi.org/10.3390/su13084259.

[12] Zhou, J., Y. Wang, F. Xiao, Y. Wang and L. Sun (2018). Water Quality Prediction Method Based on IGRA and LSTM. *Water*, MDPI, 10(9). https://doi.org/10.3390/w10091148.

[13] Zhang, Y., P. Thorburn, M. Vilas and P. Fitch (2019). Machine learning approaches to improve and predict water quality data. https://doi.org/10.36334/MODSIM.2019.D5.ZHANGYIF.

[14] Hastie, T., R. Tibshirani, J.H. Friedman and J.H. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Nature. https://doi.org/10.1007/978-0-387-21606-5.

[15] Idroes, G.M., T.R. Noviandy, A. Maulana, Z. Zahriah, S. Suhendrayatna, E. Suhartono, K. Khairan, F. Kusumo, Z. Helwani and S. Abd Rahman (2023). Urban air quality classification using machine learning approach to enhance environmental monitoring. *Leuser Journal of Environmental Studies*, Heca Sentra Analitika, 1(2), pp. 62–68. https://doi.org/10.60084/ljes.v1i2.99.

[16] Sasmita, N.R., S. Ramadeska, Z.M. Kesuma, T.R. Noviandy, A. Maulana, M. Khairul and R. Suhendra (2024). Decision Tree versus k-NN: A Performance Comparison for Air Quality Classification in Indonesia. *Infolitika Journal of Data Science*, Heca Sentra Analitika, 2(1), pp. 9–16. https://doi.org/10.60084/ijds.v2i1.179.

[17] Putra, F.M. and I.S. Sitanggang (2020). Classification model of air quality in Jakarta using decision tree algorithm based on air pollutant standard index, In *IOP Conf Ser Earth Environ Sci, IOP Publishing*, Purpose-led Publishing, p. 12053. DOI: 10.1088/1755-1315/528/1/012053

[18] Saxena, A. and S. Shekhawat (2017). Ambient air quality classification by grey wolf optimizer-based support vector machine. *Journal of Environmental and Public Health*, Wiley Online Library, 2017(1), pp. 3131083. https://doi.org/10.1155/2017/3131083.

[19]https://www.kaggle.com/datasets/supriyoain/water-quality-data.

[20] Vapnik, V (1998). *Statistical Learning Theory. New York*. John Willey & Sons. Inc.

[21] Maldonado, S., J. Pérez, R. Weber and M. Labbé (2014). Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, Elsevier, 279, pp. 163–175. https://doi.org/10.1016/j.ins.2014.03.110.

[22] Geurts, P., D. Ernst and L. Wehenkel (2006). Extremely randomized trees. *Machine Learning*, Springer Nature, 63, pp. 3–42. https://doi.org/10.1007/s10994-006-6226-1.

[23] Talatahari, S. and M. Azizi (2021). Chaos game optimization: a novel metaheuristic algorithm. *Artificial Intelligence Review*, Springer Nature, 54(2), pp. 917–1004. https://doi.org/10.1007/s10462-020-09867-w.

[24] Hippke, M. and R. Heller (2019). Optimized transit detection algorithm to search for periodic transits of small planets. *Astronomy & Astrophysics*, EDP Sciences, 623, p. A39. https://doi.org/10.1051/0004-6361/201834672.

[25]https://medium.com/@impythonprogrammer/evaluation-metrics-for-classification-fc770511052d#:~:text=Accuracy.