Water Classification Based On Mineral Content by Psobpnn

Pushparani S[1], Vallinayagam V[2]

[1]Meenakshi college of Engineering, Chennai, India
[2]St Joseph's college of Engineering, Chennai, India

Received: 11.07.2024 Revised: 16.08.2024 Accepted: 24.09.2024

ABSTRACT
Water is the very essence and the most invaluable natural resource. Over the recent decades, the degradation of water quality has been notable, primarily attributed to pollution and various other challenges. This has created a pressing demand for a model capable of providing precise forecasts regarding water quality. This research paper introduces a novel methodology for water classification based on mineral content, leveraging the Particle Swarm Optimization-enhanced Back propagation Neural Network (PSOBPNN). The study focuses on accurately categorizing water samples into distinct classes by analyzing their mineral composition, aiming to contribute to the field of water quality assessment. The proposed PSOBPNN model is employed to effectively learn and discern patterns in the mineral composition of water samples. The integration of Particle Swarm Optimization with the Back propagation Neural Network enhances the model's optimization capabilities, facilitating accurate and efficient convergence to optimal solutions. The experimental results showcase the effectiveness of the PSOBPNN model in achieving a high level of accuracy in water classification based on mineral content. The study underscores the potential significance of this approach in environmental monitoring, emphasizing the importance of considering mineral composition as a key determinant of water quality.

Keywords: Particle Swarm optimization, Back propagation, Neural Network, Mineral, Water quality

1. INTRODUCTION

Water, an indispensable resource upon which all life depends. The pollution of water degrades its quality, posing a threat to the well-being of marine organisms and consequently affecting humans who depend on them. Hence, it is imperative to monitor water quality diligently to safeguard the survival of aquatic life [1]. The availability of water holds a pivotal role in human existence and is currently recognized as a fundamental human entitlement. Goal 6 of the UN's 2015 SDGs, aiming for inclusive well-being, prioritizes ensuring everyone has safe drinking water[2]. Water quality pertains to the chemical, physical, and biological attributes of water, evaluated against established standards for its intended use [3][4]. Typically, it is gauged against a set of criteria to evaluate compliance, often achieved through water treatment. The prevalent standards for monitoring water quality serve as indicators of ecosystem health, safety for human interaction, the prevalence of water pollution, and the state of drinking water. The quality of water significantly influences water supply and frequently dictates the available supply alternatives [5].

Various water sources, such as rivers, streams, rain, and groundwater accessed through wells and boreholes, serve as crucial providers for both drinking and irrigation purposes. The characteristics of these water sources play a pivotal role in determining the composition of water samples gathered from them. Apart from natural influences, human activities like mining, crude oil extraction, and industrial discharges often introduce chemical pollutants into streams and rivers, altering the properties of the water.

As these modified waters make their way into households or farms, they become integral to domestic activities, livestock hydration, and crop irrigation. Consumption of such contaminated water can lead to severe health repercussions, including fatal outcomes. Therefore, establishing a comprehensive monitoring process is essential to track the water quality from its origin to its final use. At each monitoring point along this journey, water samples must be collected and analyzed to evaluate their suitability for human and animal consumption, as well as for irrigation and various domestic or industrial applications.

Water quality assessment is a critical aspect of environmental monitoring, as it directly influences the health of ecosystems and the well-being of human populations [6]. The increasing anthropogenic activities, urbanization, and industrialization have heightened concerns about water pollution and the

need for efficient methodologies to analyze and manage water quality[7]. In this context, the integration of advanced computational techniques with environmental science has emerged as a promising avenue for accurate and timely water quality monitoring.

This research article explores the application of a novel methodology, combining Particle Swarm Optimization (PSO) and Neural Networks, to address the challenges in water quality analysis. The integration of PSO, a nature-inspired optimization algorithm, with a neural network model enhances the accuracy and efficiency of water quality prediction, providing a powerful tool for environmental researchers and policymakers.

Traditional water quality monitoring methods often face limitations in terms of real-time data processing, adaptability to dynamic environmental changes, and the ability to handle complex datasets. The proposed PSO Neural Network approach aims to overcome these challenges by leveraging the optimization capabilities of PSO to enhance the training and fine-tuning processes of neural networks. This synergy offers a more robust and adaptive system for predicting water quality parameters, leading to improved decision-making in sustainable resource management.

Throughout this article, we delve into the theoretical underpinnings of the PSO Neural Network model and showcase its application in real-world water quality datasets. We assess the model's performance against established benchmarks, highlighting its advantages in terms of accuracy, speed, and adaptability. Furthermore, we discuss the implications of our findings for water resource management, environmental policy, and the broader field of data-driven environmental science.

The integration of PSO and Neural Networks represents a promising approach for advancing the field of water quality analysis. This research contributes to the ongoing efforts to develop innovative and effective tools for sustainable resource management, with the potential to positively impact both environmental conservation and human well-being.

2. RELATED WORKS

The prediction of river water quality has seen a great deal of recent investigation and use of machine learning technology [8]. When it comes to river water quality predictions, machine learning uses a wealth of historical data to create accurate prediction models that allow early warning systems. There are many benefits associated with this technique [9]. The main benefit of this is that it makes it possible to monitor and anticipate water quality in real-time and continuously, which improves the effectiveness and responsiveness of water quality management. Second, machine learning algorithms may learn and adapt on their own to complex relationships seen in water quality data, which can result in forecasts that are more accurate [10]. These models can also incorporate meteorological data and other environmental elements, improving the precision and dependability of water quality forecasts.

Despite the benefits, machine learning systems encounter challenges and limitations in predicting river water quality. Factors like data quality and missing data can affect the performance of the model [11]. Additionally, the training process and parameter selection demand a certain level of expertise and experience. Moreover, the model's interpretability is insufficient, making it hard to comprehend the forecasted results. Consequently, further research and development efforts are necessary to enhance the effectiveness and reliability of machine learning in predicting river water quality.

An information-theory-based technique used to assess the information's and significance of characteristics is the entropy weighting method [12]. The purity and discriminability of these features can be evaluated by calculating their feature entropy values [13]. By combining the entropy weighting approach with the Pearson correlation coefficient, one can reduce dependence on a single feature selection criterion by taking into account both correlation and information content. By eliminating subjectivity and uncertainty, this integrated approach makes it easier to evaluate feature contribution and importance in greater detail. Finding a balance between many parameters makes it possible to choose features that have higher correlation and more information, which improves feature selection accuracy and stability.

Table 1. Comparative evaluation of the classification methods employed in current studies

Data size	Method	Accuracy (%)	Reference
370	ARIMA	74	18
1912	PNN	82	20
273	SVR	77	27
896	ANFIS	81	28

Both the entropy weighting method and the Pearson correlation coefficient are very simple and obvious procedures that are easy to understand [14]. When they are used in machine learning feature selection,

the results are more practical and easy to understand. This integration helps decision-makers understand the significance and contribution of features by improving the transparency and dependability of the feature-selection process. It enhances the interpretability and efficiency of the model at the same time [15].Top of Form

Human activity in urban areas is a major culprit behind polluted water, with municipal and industrial wastewater being the main offenders [16]. This has led to a surge in research on using machine learning to predict and analyze surface water quality [17, 18]. As a result, various methods have been developed. Researchers are actively working to fine-tune these models and enhance their accuracy.

Several researchers have investigated machine learning techniques for predicting water quality. Yafra Khan and Chai Soo See developed a model that combines Artificial Neural Networks with time series analysis, assessing its performance using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Regression Analysis [19]. Dao Nguyen Khoi et al. utilized 12 different machine learning models and evaluated them using metrics like R-squared (R2) and RMSE [20]. Umair Ahmed et al. employed supervised learning to estimate the Water Quality Index (WQI) [21]. Saber Kouadri et al. compared eight AI algorithms for WQI prediction, evaluating them with metrics such as R-squared (R), Mean Absolute Error (MAE), RMSE, Relative Absolute Error (RAE), and Relative Root Mean Squared Error (RRSE) [22]. Additionally, Jitha Nair and Vijaya MS explored various prediction models based on machine learning and big data techniques using sensor networks [23]. The application of machine learning in surface water quality research has garnered significant attention [28, 29]. Various methods have been devised for predicting and analyzing surface water quality. Considerable efforts have been devoted to refining machine learning models and improving their predictive accuracy.

Previous research on water quality prediction utilized various machine learning models such as XGBoost, Random Forest, Decision Tree, AdaBoost, and SVC. Among these, XGBoost achieved the best performance with an accuracy of 83%.[24]

Table 2. Data description

S.No	Parameters	Acceptable Limits
1	aluminium	2.8
2	ammonia	32.5
3	arsenic	0.01
4	barium	2
5	cadmium	0.005
6	chloramines	4
7	chromium	0.1
8	copper	1.3
9	fluoride	1.5
10	bacteria	0
11	viruses	0
12	lead	0.015
13	nitrates	10
14	nitrites	1
15	mercury	0.002
16	perchlorate	56
17	radium	5
18	selenium	0.5
19	silver	0.1
20	uranium	0.3

3. ARTIFICIAL NEURAL NETWORKS (ANNs):

Artificial Neural Networks (ANNs) are one of the bio inspired computational models inspired by the structure and functioning of the neurons in the human brain. It is a mathematical model composed of interconnected nodes, called neurons or perceptrons, organized into layers. ANNs are capable of learning from data to approximate complex functions and make predictions or classifications. They are widely used in machine learning for tasks such as pattern recognition, classification, regression, and optimization.

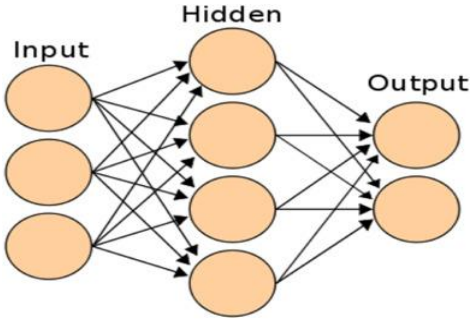


Figure 1. Artificial Neuron

Neurons are the basic computational units in ANNs. Each neuron receives input signals, processes them using a defined activation function, and produces an output signal. In a typical ANN, neurons are organized into layers: input, hidden, and output layers.

ANNs have three major types of layers: Input Layer: This layer gets the first input data. The number of neurons in this layer corresponds to the number of inputs. Hidden Layer(s): These layers process and transform input via weighted connections between neurons. The formula(1) defines the number of neurons in this layer.

$$E = \frac{1}{2}\sum_{k=1}^{n_3}\left(y_k - f_1\left(\sum_{j=1}^{n_2} V_{jk} f_0\left(\sum_{i=1}^{n_1} a_i w_{ij} - \theta_j \right) - \eta_k \right) \right) \quad (1)$$

In this equation, η_k represents the threshold for the output layer, while θj denotes the threshold of the hidden layer. The connection weight from the input layer to the hidden layer is denoted as W_{ij}, and the connection weight from the hidden layer to the output layer is represented by V_{jk}. The activation function f_0 corresponds to the Sigmoid function used in the hidden layer, while f_1 refers to the linear function used in the output layer.

Output Layer: This layer generates the final output or forecast. The number of neurons in this layer is determined by the no of categories in the data set. Figure 1 [27] depicts a visual representation of the applied Back Propagation Network.

Connections between neurons in adjacent layers are established via weights, which dictate the connections' potency and are fine-tuned throughout the learning phase. Learning entails modifying these weights according to the network's performance on training data. The activation function governs a neuron's output by processing its weighted inputs. Typical activation functions comprise sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). The selection of activation function impacts the network's capacity to grasp intricate relationships.

Artificial Neural Networks (ANNs) acquire knowledge from data using a technique known as backpropagation. This iterative process aims to minimize the disparity between the predicted output and the desired target values by adjusting the network's weights. Within an ANN, each neuron is fed input signals (x_1, x_2, ..., x_n), each multiplied by a respective weight (w_1, w_2, ..., w_n). The resulting weighted sum of inputs undergoes processing through an activation function, represented as 'a' (e.g., sigmoid, tanh, ReLU).

Mathematically, the output y of a neuron is calculated as

$$y = a(\sum_{i=1}^{n} w_i . x_i + b) \quad (2)$$

where b is the bias term.

In a feed forward neural network, the output from one layer acts as the input to the next layer. This progression through the network can be represented mathematically using matrix multiplication. Let X denote the input vector, W represent the weight matrix, b indicate the bias vector, and A signify the output after applying the activation function. The feedforward process for a layer is described by the equation:

$$A = a(W \cdot X + b) \quad (3)$$

This operation is iterated for every layer within the network, where the output from one layer becomes the input for the subsequent layer.

The final layer in the network produces the output. For a classification task, the output is often passed through a soft max function to convert the raw scores into probabilities.

Let Z be the output of the final layer before the softmax, and Y be the final output. The softmax function is defined as:

$$Y_i = \frac{e^{Z_i}}{\sum_{j=1}^{c} e^{Z_j}} \quad (4)$$

where z_i is the input to the softmax function corresponding to class i, and K is the total number of classes. The function computes the probability distribution over multiple classes, ensuring that the output values are non-negative and sum up to 1.

A loss function measures the difference between the projected output and the actual target. Mean squared error and cross-entropy are two common loss functions for regression and classification applications, respectively.

In a classification problem, the cross-entropy loss between predicted probability distribution Y and true distribution T is defined as:

$$L(Y,T) = -\sum_i T_i.\log(Y_i) \quad (5)$$

Backpropagation is the algorithm utilized to adjust the weights of the network to minimize the loss. It computes the gradients of the loss concerning the weights using the chain rule of calculus, which are then utilized to update the weights through optimization algorithms such as gradient descent.

The weight update for a given weight w is typically performed as:

$$w_{new} = w_{old} - \alpha.\frac{\partial L}{\partial w} \quad (6)$$

Where α is the learning rate.

4. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a meta-heuristic algorithm widely utilized for addressing discrete, continuous, and combinatorial optimization challenges. It was initially introduced by Kennedy and Eberhart in 2001 [25], drawing inspiration from the flight patterns of bird flocks. In the context of PSO, an individual solution is termed a particle, while the aggregation of all solutions is known as a swarm. The core concept underlying PSO is that each particle possesses knowledge of its current velocity, its own best solution achieved in the past (pBest($\breve{x}_t^n$)), and the global best solution currently observed by the entire swarm (gBest($\breve{g}_t^n$)). During each iteration, every particle adjusts its velocity to steer closer to both its pBest and gBest [26]. The velocity of each particle, denoted as v, is adapted according to the following equation:

that its new position is closer to both its pBest and gBest.[26]. Each particle's velocity, *v*, is adjusted in accordance with the subsequent equation:

(7)

$$v_{t+1}^n = wv_t^n + \rho_1 r_1(\breve{g}_t^n - x_t^n) + \rho_2 r_2(\breve{x}_t^n B - x_t^n)$$

In the given equation, v represents the particle's velocity, x denotes its current position, w is a constant known as momentum that regulates the influence of the velocity from the previous time step on the current velocity, ρ_1 and ρ_2 are predefined constants, and r_1 and r_2 are random numbers in the range [0, 1]. Subsequently, the position of the i^{th} particle is updated as follows

$$x_{t+1}^n = x_t^n + v_{t+1}^n \quad (8)$$

PSO is a stochastic optimization algorithm in which a group of particles collectively explores the solution space by adjusting their positions and velocities. This adjustment is guided by the best positions discovered by each particle individually, as well as those found by their neighboring particles. Through this swarm-based approach, PSO efficiently traverses complex solution spaces in pursuit of optimal solutions

The Particle Swarm Optimization (PSO) algorithm, when integrated with a neural network, serves the purpose of optimizing the weights within the neural network architecture. This optimization process aims to enhance the neural network's capability to predict water quality accurately, utilizing provided data.

5. METHODOLOGY

Figure 2 illustrates the complete workflow of the proposed system, detailing the steps from gathering data to the training and testing stages, incorporating custom models as well as others. The dataset was split into training and testing sets to aid in the development and assessment of models.

Water has been classified using a number of cutting-edge classification models in recent years, mostly using statistical techniques. In order to classify water quality from CSV datasets, this study includes five pre-trained models and a unique PSO Neural Network model. This model includes nature inspired Particle Swarm Optimization for optimize the weight and bias of an Back propagation Artificial Neural Network to improve the accuracy of the classifier

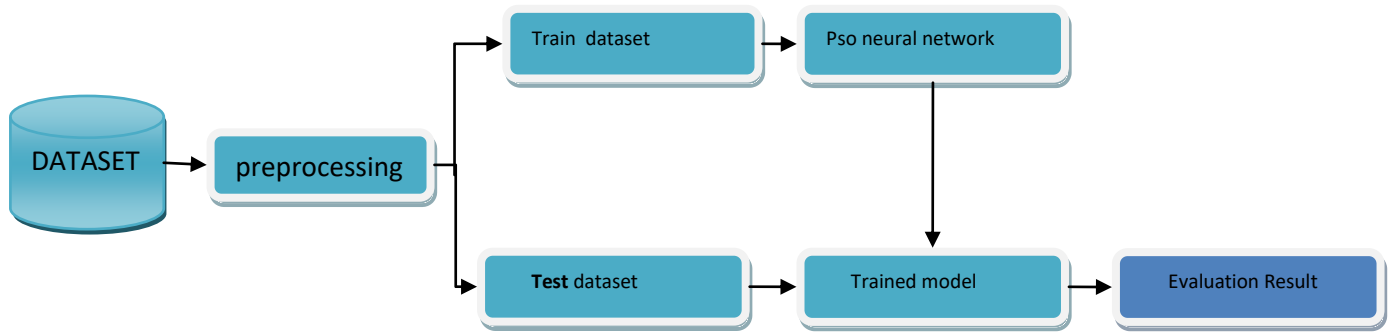


Figure 2. Analysis Model

6. PSOBPANN Model

The proposed PSOBPANN model consists of three layers with softmax as the activation function. PSO is employed to optimize the weights and biases of a back propagation neural network because of its stochastic nature, which helps prevent premature convergence to suboptimal solutions. The random movements of particles enable the algorithm to escape local optima and explore a broader solution space.
A particle in the PSO-BP algorithm is represented by a set of parameters, specifically the weights (W) and biases (B) of the neural network. Let x_i represent the position of particle i, where

$$x_i = [w_{i1}, w_{i2}, ..., w_{ij}, b_{i1}, b_{i2}, ..., b_{ik}]. \quad (9)$$

Here the neural network consists of an input layer, a single hidden layer, and an output layer. Let X be the input data, H be the hidden layer output, and Y be the network output.
A sigmoid activation function is applied to the output of each neuron in the output layer:

$$f(x) = \frac{1}{1+e^{x}} \quad (10)$$

The hidden layer output (H) and the network output (Y) are calculated using the current particle's position (x_i) as

$$H = \sigma(X.W_{hi} + B_{hi}) \quad (11)$$

$$H = \sigma(H.W_{oh} + B_{oh}) \quad (12)$$

where σ is the sigmoid activation function.
The objective function to be minimized is the error function of the neural network is

$$E(x_i) = \frac{1}{2}\sum_{j=1}^{N}(D_j - Y_j)^2 \quad (13)$$

The gradients of the objective function with respect to the weights(W) and biases(B) are computed using back propagation. Update the weights and biases are updated using gradient descent algorithm:

$$W_{hi} = W_{hi} - \eta\frac{\partial E}{\partial W_{hi}} \quad (14)$$

$$B_{hi} = B_{hi} - \eta\frac{\partial E}{\partial B_{hi}} \quad (15)$$

$$W_{oh} = W_{oh} - \eta\frac{\partial E}{\partial W_{oh}} \quad (16)$$

$$B_{oh} = B_{oh} - \eta \frac{\partial E}{\partial B_{oh}} \quad (17)$$

where η is the learning rate.
particle positions are updated using the PSO equations (7) and (8)
After updating the particle position, the sigmoid activation function is applied to the hidden layer output again as

$$H = \sigma(X.W_{hi} + B_{hi}) \quad (18)$$

the fitness of the particle is evaluated using the updated position and the error function by

$$E(x_i) = \frac{1}{2}\sum_{j=1}^{N}(D_j - Y_j)^2 \quad (19)$$

personal best positions(p_i) and global best position (p_g) are updated based on the current fitness.
PSO is used to find the optimal weights for a neural network that can effectively predict water quality based on the given data. The PSO algorithm guides the search for the best set of weights in the solution space, and the neural network's performance is continuously improved throughout the iterations. The final trained neural network, with the optimized weights, is then evaluated on new data.

7. RESULTS AND DISCUSSION

This section provides a comprehensive examination of the dataset, experimental procedures, model training, and validation processes. Additionally, it presents a thorough performance comparison between the proposed technique and previous approaches

a. Experimental Setup.

The implementation of the proposed work was carried out in Google Colab notebooks, a cloud computing environment. Google Colab provides access to a free tensor processing unit (TPU) and graphics processing unit (GPU) for the development of neural learning models. The custom PSOBPNN was coded using the Python programming language.

b. Data set

This approach utilized data sourced from Kaggle's Water Quality Dataset, encompassing various metrics such as aluminium, ammonia, arsenic, barium, cadmium, chloramines, chromium, copper, fluoride, bacteria, viruses, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, and uranium. The dataset comprises 7996 data points, with 7084 falling under the 'not safe' category and 912 classified as 'safe'. Sample feature descriptions are outlined in Table 2, while Table 3 illustrates sample data.

Table 3. Partial sample data

lead	nitrates	nitrites	mercury	perchlorate	radium	selenium	silver	uranium	is_safe
0.054	16.08	1.13	0.007	37.75	6.78	0.08	0.34	0.02	1
0.1	2.01	1.93	0.003	32.26	3.21	0.08	0.27	0.05	1
0.078	14.16	1.11	0.006	50.28	7.07	0.07	0.44	0.01	0
0.016	1.41	1.29	0.004	9.12	1.72	0.02	0.45	0.05	1
0.117	6.74	1.11	0.003	16.9	2.41	0.02	0.06	0.02	1
0.135	9.75	1.89	0.006	27.17	5.42	0.08	0.19	0.02	1
0.021	18.6	1.78	0.007	45.34	2.84	0.1	0.24	0.08	0

c. Data preprocessing

The missing values in the data set are handled by Imputation, a statistical technique used to replace missing values in a dataset with estimated values. There are various imputation methods, and one common approach is to replace missing values with the mean of the observed values in the variable. Here's an explanation of handling missing data by mean imputation:

The mean $\overline{X}$ of the observed values in the variable with missing data is calculated using.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} \quad (20)$$

Where X_i is the observed value, and n is the number of observed values. Then the missing values were replaced with calculated mean of the observations.

Preprocessing involves performing statistical calculations on the attributes of a dataset. These calculations encompass metrics like mean, standard deviation, minimum, maximum, and quartiles, offering insights into the distribution and characteristics of the data shown in Table 4

The analysis involved examining the correlation matrix of the dataset features, as illustrated in Figure 3. This matrix delves into the connections between various features, aiding in the identification of noteworthy associations or dependencies among the variables.

Table 4: Statistical calculation of the features

	aluminium	arsenic	barium	cadmium	chloramine	chromium	copper	flouride	bacteria	viruses	lead	nitrates	nitrites	mercury	perchlorate	radium	selenium	silver	uranium	
count	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000	7999.000000
mean	0.666158	0.161445	1.567715	0.042806	2.176831	0.247226	0.805857	0.771565	0.319665	0.328583	0.099450	9.818822	1.329961	0.005194	16.460299	2.920548	0.049685	0.147781	0.044673	
std	1.265145	0.252590	1.216091	0.036049	2.567027	0.270640	0.653539	0.435373	0.329485	0.378096	0.058172	5.541331	0.573219	0.002967	17.687474	2.323009	0.028770	0.143551	0.026904	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.040000	0.030000	0.560000	0.008000	0.100000	0.050000	0.090000	0.405000	0.000000	0.000020	0.004800	5.000000	1.000000	0.003000	2.170000	0.820000	0.020000	0.040000	0.020000	
50%	0.070000	0.050000	1.190000	0.040000	0.530000	0.090000	0.750000	0.770000	0.220000	0.000800	0.102000	9.930000	1.420000	0.005000	7.740000	2.410000	0.050000	0.080000	0.050000	
75%	0.280000	0.100000	2.480000	0.070000	4.240000	0.440000	1.390000	1.160000	0.610000	0.700000	0.151000	14.610000	1.760000	0.008000	29.480000	4.670000	0.070000	0.240000	0.070000	
ma	5.05	1.05	4.94	0.13	8.68	0.90	2.00	1.50	1.00	1.00	0.20	19.8	2.93	0.01	60.0	7.99	0.10	0.50	0.09	

aluminium	arsenic	barium	cadmium	chloramine	chromium	copper	flouride	bacteria	viruses	lead	nitrates	nitrites	mercury	perchlorate	radium	selenium	silver	uranium	
x	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	30000	0000	0000	10000	0000	0000	0000	0000

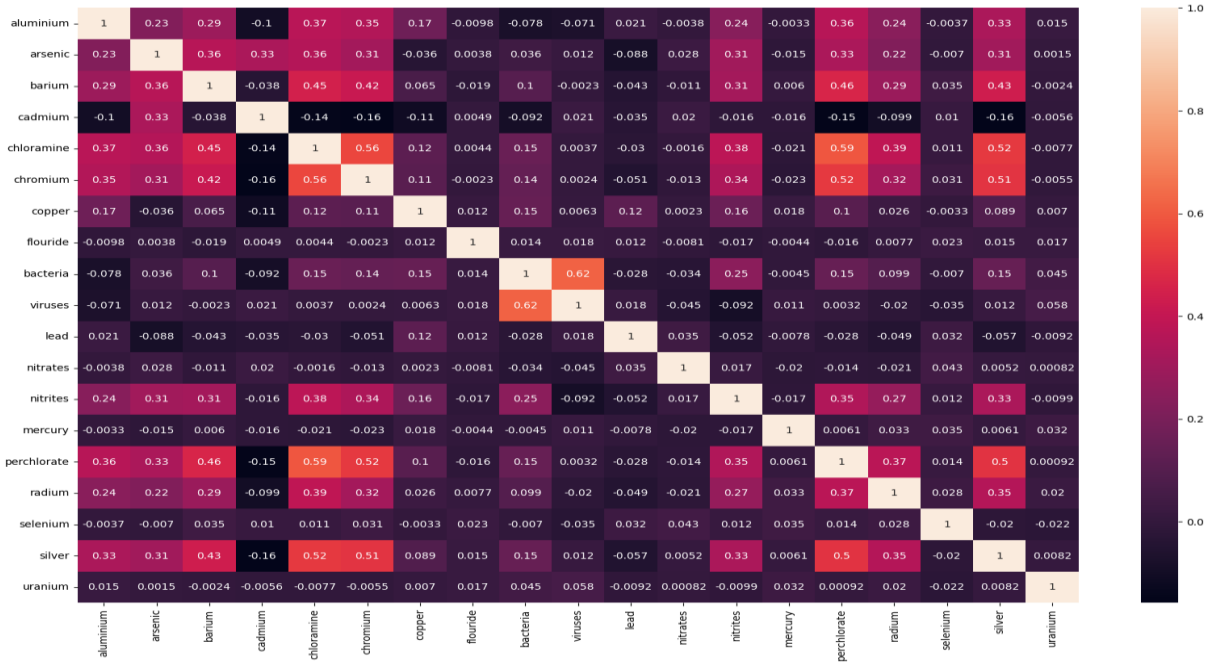


Figure 3. Visualization of feature correlations

d. Data splitting

The imputed data is randomly divided into distinct subsets for training and testing. The training dataset comprises 80% of the overall dataset, while the testing dataset constitutes the remaining 20%. In the process of building a machine learning model, a connection is established between independent and dependent parameters to make predictions or select alternatives. Subsequently, the effectiveness of the machine learning technique is assessed using the test data to determine its performance.

Table 5: Algorithm parameters

Particle size	20
No of iterations	100
Swarm size	20
C1	1.5
C2	2.5
No of neurons in input layer	20
No of neurons in hidden layer	5
Epochs for evaluation	1000

e. Hyper parameter optimization

Grid search was employed to optimize hyperparameters in this model, specifically focusing on adjusting the inertia weight within the PSO (Particle Swarm Optimization) algorithm. These hyperparameters

influence the behavior of the PSO algorithm and can have a significant impact on its convergence and exploration-exploitation trade-off.

f. Performance metrics

Evaluation metrics are pivotal in determining the efficacy of a trained model. The PSOBPNN model's effectiveness was assessed using several key metrics, including precision, accuracy, F1-score, recall, and the analysis of a confusion matrix. The accuracy during testing was computed by predicting outcomes using the trained model on a designated test set. The confusion matrix provided insights into the model's performance across various classes. Testing accuracy was obtained by evaluating the model's predictions on the test set, which was segregated during the dataset partitioning phase. Equations (21) to (24) encapsulate the mathematical formulations for these evaluation metrics.

$$Accuracy = \frac{A+B}{A+B+C+D} \quad (21)$$

$$\Pr ecision = \frac{A}{A+B} \quad (22)$$

$$\mathrm{Re}call = \frac{A}{A+D} \quad (23)$$

$$F1Score = \frac{2*\Pr ecision*\mathrm{Re}call}{\Pr ecision+\mathrm{Re}call} \quad (24)$$

A denotes instances correctly classified as the positive class, while B represents instances correctly classified as the negative class. C signifies instances incorrectly classified as the positive class, and D denotes instances incorrectly classified as the negative class. True Positives (A) arise when the model accurately predicts the positive class, while True Negatives (B) occur when the model accurately predicts the negative class. False Positives (C) occur when the model erroneously predicts the positive class as negative, and False Negatives (D) happen when the model incorrectly predicts the negative class as positive.

i. water classification

Traditionally, water quality assessment relies on time-consuming laboratory analysis to obtain water quality criteria. In our study, diverse machine learning method for estimating water quality, drawing insights from existing research that employed these approaches. The model's performance was assessed based on mineral content of water. The following algorithm is used for implementing the model for water classification based on mineral content

Algorithm 1: Particle Swarm Optimization (PSO) for Water Quality Classification

```
{
        Load the dataset
        Preprocess the dataset
        Split the dataset into training and testing sets
        Set PSO parameters
            Initialize particles
                Initialize global best position and score
# PSO algorithm
        Initialize particles
For each particle:
  Initialize position and velocity randomly
  Evaluate fitness of the particle (accuracy of the neural network with current weights)
 Initialize global best position and score
For each iteration:
  For each particle:
    Update particle's velocity and position
    Train neural network with current particle's position
    Evaluate fitness of the particle (accuracy of the neural network with current weights)
    Update particle's best position and global best position
End of PSO iterations
 Train final neural network with the global best position
 Evaluate the performance of the final model on the test set
```

Train final neural network with the global best position
Evaluate the performance of the final model on the test set
}

In the aforementioned algorithm, the parameters detailed in table 5 are employed to regulate the behavior of the PSO algorithm. The number of iterations determines the total iterations the algorithm will execute before deeming the optimization complete. The swarm size dictates the quantity of particles utilized by the PSO algorithm. The parameter Cg plays a pivotal role in governing the speed at which particles in the swarm converge towards the global best particle. The number of input neurons is contingent upon the variables present in the dataset.

j. Performance Analysis

By using Particle swarm optimization for optimizing Back propagation neural network for water analysis improves the classification accuracy .The confusion matrix for this work is given in table 6. PSOBPNN classifies 1369 data sets as True positive, 162 as true negative, 31 data sets as false-positive and 38 data sets as false negative which is close to 95.60% classification accuracy. Comparison of results shown in Figure 4.

The results obtained from experiments comparing the performance of PSOBPNN against Random Forest (RFT), Decision Tree, Support Vector Classifier, K Nearest Neighbor Classifier, and Feed forward Neural Network Classifier highlight the effectiveness of PSOBPNN. The PSOBPNN model demonstrated superior performance compared to other classification models, achieving high recall and precision values. The values are computed from corresponding confusion matrix in Table 6, by using the equations (21) to (24)

k. Comparison with other methods for water classification

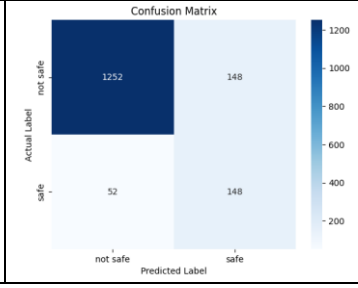
Table 1 compares the results of the PSOBPNN model with the other methods assessed on other datasets in recent years. In Ref [8], Autoregressive integrated moving average (ARIMA) models is used can accommodate less number of data and gives the accuracy 74%. Khoi etal[20] achieved accuracy of 82% by using Bayesian based Probabilistic neural network for water quality prediction, Tang, et al[27] employed Support Vector Regression and produced the accuracy of 77%. Tung et al[28] used an adaptive neuro-fuzzy inference system that predicts with 81% accuracy. Overall, the PSOBPNN model achieved the highest accuracy (95.6%) and performed better than other methods.

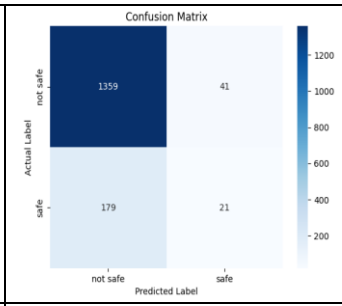
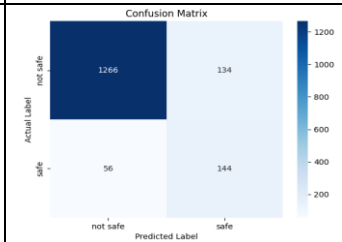
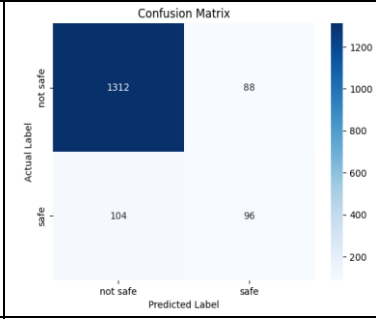
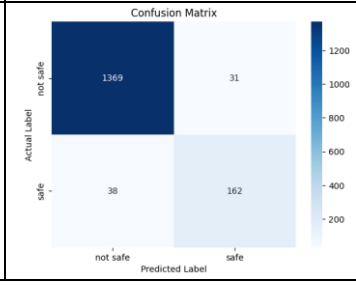
l. Comparison with existing methods

To assess the effectiveness of PSOBPNN classifier, we compared the performance metrics of various classifiers such as Random Forest (RF), Decision Tree, Support Vector Classifier, K-Nearest Neighbor classifier, and feed forward neural network classifier. We generated confusion matrices and classification reports for each classifier. Table 6 displays the confusion matrices and Table 7 shows the classification report.

On this particular data set, PSOBPNN classifies with higher accuracy as compare with other methods used with the same data set. Figure 5 shows the accuracy and loss curves for both the training and validation sets. The training loss curve starts off high and then decreases rapidly. This suggests that the model is quickly learning the training data. However, the training loss starts to increase again after about 10 epochs. The validation loss curve starts off high and then decreases slowly. This suggests that the model is slowly learning the training data. The training accuracy curve starts off low and then increases rapidly. This suggests that the model is quickly learning to classify the training data correctly. The validation accuracy curve starts off low and then increases slowly. This suggests that the model is slowly learning to classify the validation data correctly.

Table 6. confusion matrices of models on the data set

Model	Confusion matrix
RANDOM FOREST	

KNN	
FFNN	
SVM	
PSOBPNN	

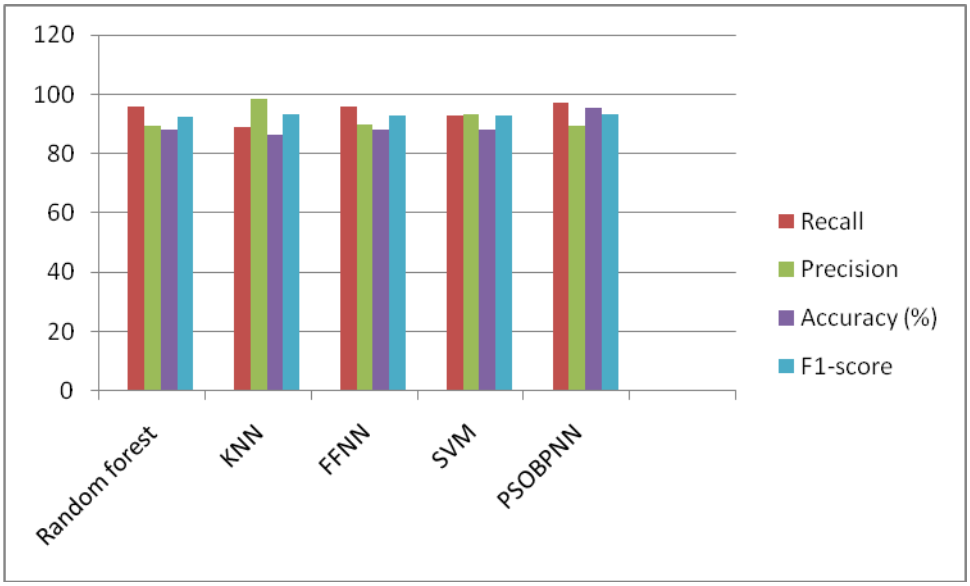


Figure 4: The comparative metric of all classifiers

Table 7. Comparison of PSOBPNN with other models on the dataset

Model	Recall	Precision	Accuracy (%)	F1-score
Random forest	96	89.42	88	92.5
KNN	88.9	98.47	86.25	93.4
FFNN	95.76	89.8	88.13	92.7
SVM	92.65	93.2	88	93
PSOBPNN	97.29	89.4	95.6	93.2

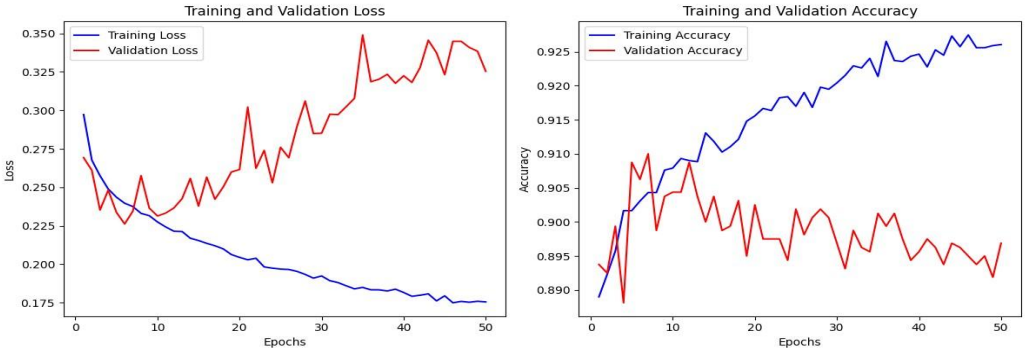


Figure 5. The learning process of the PSOBPNN model is visualized through curves depicting the epochs, model loss, and model accuracy

CONCLUSION

This work has demonstrated the effectiveness of the Particle Swarm Optimization-enhanced Backpropagation Neural Network (PSOBPNN) in accurately classifying water samples based on their mineral content. The achieved accuracy of 95% reflects the robustness and reliability of the proposed model in distinguishing between different water classifications. The integration of PSO optimization with the backpropagation neural network has proven to enhance the model's ability to converge to optimal solutions and improve overall classification performance.

The findings of this research contribute to the growing body of knowledge on water quality assessment, offering a reliable and accurate method for classifying water samples based on heavy metal content. The potential societal impact of such a model in safeguarding water resources underscores the significance of continued exploration and refinement of advanced computational techniques for environmental monitoring.

In this study, the dataset comprises information on mineral content in water, for the classification of water as either safe or not safe. However, it is important to note that the absence of additional detailed features, such as Biological Oxygen Demand (BOD) and pH levels, may limit the classifier's predictive accuracy. Including these additional features in the dataset has the potential to significantly enhance the model's performance by providing a more comprehensive understanding of water quality parameters.

Incorporating BOD and pH, among other relevant features, into the analysis could contribute valuable insights into the overall water quality assessment. BOD is a crucial parameter indicating the amount of dissolved oxygen required by microorganisms to break down organic matter in water, while pH levels offer insights into the acidity or alkalinity of the water. The inclusion of such features could offer a more holistic view of water quality, enabling the classifier to make more informed and accurate predictions.

Future research endeavors may consider expanding the dataset to include a broader range of water quality parameters, thereby improving the model's ability to discern safe and unsafe water classifications. Additionally, exploring the synergies between heavy metal content and other features could lead to a more robust and reliable classification model, with potential applications in water resource management and public health.

REFERENCES

[1] Jain D, Shah S, Mehta H et al (2021) A Machine Learning Approach to Analyze Marine Life Sustainability. In: Proceedings of International Conference on Intelligent Computing, Information and Control Systems. Springer, pp 619–632

[2] B. X. Lee, F. Kjaerulf, S. Turner, L. Cohen, P. D. Donnelly, R. Muggah, R. Davis, A. Realini, B. Kieselbach, L. S. Mac Gregor, I.Waller, R. Gordon, M. Moloney-Kitts, G. Lee, and J. Gilligan, “Transforming our

world: Implementing the 2030 agenda through sustainable development goal indicators, " J. Public Health Policy, vol. 37, no. S1, pp. 13_31, Sep. 2016.
[3] Cordy, Gail E. (March 2001). "A Primer on Water Quality". Reston, VA: U.S. Geological Survey (USGS). FS-027-01
[4] Johnson D. L, Ambrose S. H, Bassett T.J, Bowen M. L, Crummey D.E, Isaacson J.S, Johnson D.N, Lamb P, Saul M, Winter-Nelson A.E. (1997). " Meanings of Environmental Terms". Journal of Environmental Quality. 26 (3):581–589. doi: 10.2134 / jeq1997. 00472425002600030002 xs.
[5] Guidelines for Drinking-water Quality: Fourth edition incorporating the first addendum (Report). Geneva: World Health Organization (WHO). 2017. hdl:10665/254637. ISBN 9789241549950.
[6] P. A. Salam, M. A. Fazal, M. Masud, "Water Quality Assessment in Terms of Water Quality Index," American Journal of Water Resources, 1(3), 34-38, 2013.
[7] D. B. Das, P. P. Kar, A. K. Mohanty, "A Machine Learning Approach for Prediction of Water Quality Index," Procedia Computer Science, 132, 157-164, 2018.
[8] Zhu, M.; Wang, J.; Yang, X.; Zhang, Y.; Zhang, L.; Ren, H.; Wu, B.; Ye, L. A review of the application of machine learning in water quality evaluation. Eco-Environ. Health 2022, 1, 10.
[9] Alizadeh, M.J.; Kavianpour, M.R.; Danesh, M.; Adolf, J.; Shamshirband, S.; Chau, K.W. Effect of river flow on the quality of estuarine and coastal waters using machine learning models. Eng. Appl. Comput. Fluid Mech. 2018, 12, 810–823.
[10] Omambia, A.; Maake, B.; Wambua, A. Water quality monitoring using IoT & machine learning. In Proceedings of the 2022 IST-Africa Conference (IST-Africa 2022), Virtual Conference, 16–20 May 2022; pp. 1–8. https://doi.10.23919/IST-Africa56635.2022.9845590.
[11] Kayhomayoon, Z.; Arya Azar, N.; Ghordoyee Milan, S.; Kardan Moghaddam, H.; Berndtsson, R. Novel approach for predicting groundwater storage loss using machine learning. J. Environ. Manag. 2021, 296, 113237.
[12] Cao, R.; Yuan, J. Selection Strategy of Vibration Feature Target under Centrifugal Pumps Cavitation. Appl. Sci. 2020, 10, 8190.
[13] Yan, X.; Liu, Y.; Jia, M. A Feature Selection Framework-Based Multiscale Morphological Analysis Algorithm for Fault Diagnosis of Rolling Element Bearing. IEEE Access 2019, 7, 123436–123452.
[14] Li, G.; Zhang, A.; Zhang, Q.; Wu, D.; Zhan, C. Pearson correlation coefficient-based performance enhancement of broad learning system for stock price prediction. IEEE Trans. Circuits Syst. II Express Briefs 2022, 69, 2413–2417.
[15] Zheng, Y.; Li, Y.; Wang, G.; Chen, Y.; Xu, Q.; Fan, J.; Cui, X. A novel hybrid algorithm for feature selection based on whale optimization algorithm. IEEE Access 2018, 7, 14908–14923.
[16] R. Mohammadpour, S. Shaharuddin, C.K. Chang, N.A. Zakaria, A. Ab Ghani, N.W. Chan, Prediction of water quality index in constructed wetlands using support vector machine, Environ. Sci. Pollut. Control Ser. 22 (2015) 6208–6219, https://doi.org/10.1007/s11356-014-3806-7.
[17] T.M. Tung, Z.M. Yaseen Tiyasha, A survey on river water quality modelling using artificial intelligence models: 2000–2020, J. Hydrol. 585 (2020), https://doi.org/ 10.1016/ j.jhydrol. 2020. 124670.
[18] N. Sharma, R. Sharma, N. Jindal, Machine learning and deep learning applications-A vision, Global Transitions Proceedings 2 (2021) 24–28, https://doi.org/10.1016/j.gltp.2021.01.004.
[19] Y. Khan and C. S. See, "Predicting and analyzing water quality using machine learning: a comprehensive model," in Proceedings of the 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), April 2016.
[20] D. N. Khoi, N. T. Quan, D. Q. Linh, P. T. T. Nhi, and N. T. D. 'uy, "Using machine learning models for predicting the water quality index in the La buong river, Vietnam," Water, vol. 14, no. 10, p. 1552, 2022.
[21] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. Garc´ıa-Nieto, "Efficient water quality prediction using supervised machine learning," Water, vol. 11, p. 2210, 2019.
[22] S. Kouadri, A. Elbeltagi, A. R. M. T. Islam, and S Kateb, "Performance of machine learning methods in predicting water quality index based on irregular data set: application on Illizi region (Algerian southeast)," Applied Water Science, vol. 11, no. 12, p. 190, 2021.
[23] J. P. Nair and M. S. Vijaya, "Predictive models for river water quality using machine learning and big data techniques – a Survey," in Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), IEEE, Coimbatore, India, March 2021.
[24] M. M. Hassan, M. M. Hassan, L. Akter et al., "Efficient prediction of water quality index (WQI) using machine learning algorithms," Human-Centric Intelligent Systems, vol. 1, no. 3-4, pp. 86–97, 2021.
[25] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Academic Press, San Diego, CA, 2001.
[26] P. J. Angeline, ``Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences,'' Evol. Program., vol. 48,no. 5, pp. 601_610, 1998.

[27] T. Tang, X. L. Zhang, H. Tao, and L. K. Wei, ``Fault diagnosis of servo mechanism for machine tool based on LM-BP neural network,'' J. Guilin Univ. Electron. Technol., vol. 33, no. 3, pp. 218_222, 2013.
[28] T.M.Tung, Z.M. Yaseen Tiyasha, A survey on river water quality modelling using artificial intelligence models: 2000–2020, J.Hydrol.585(2020), https://doi.org/10.1016/j.jhydrol.2020.124670.
[29] N. Sharma, R. Sharma, N. Jindal, Machine learning and deep learning applications-A vision, Global Transitions Proceedings 2 (2021) 24–28, https:/doi.org/10.1016/j.gltp.2021.01.004.