STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 6076–6095.
Published online in International Academic Press (www.IAPress.org)

Optimizing Deep Neural Networks for Water Quality Prediction Using Spider Monkey Optimization and Permutation Importance-Based Feature Selection: A Comparative Heuristic Approach

Desy Lusiyanti [1], Syaiful Anam [1,*], Wayan Firdaus Mahmudy [2], Umu Sa'adah [1]

[1] *Department of Mathematics, Faculty of Mathematics and Natural Science, University of Brawijaya Malang, Indonesia*
[2] *Faculty of Computer Science, University of Brawijaya Malang, Indonesia*

Abstract This study evaluates Spider Monkey Optimization (SMO) for hyperparameter optimization of a deep neural network (DNN) for water potability classification and compares its performance with Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and the Firefly Algorithm (FA) under a common hyperparameter search space, 10-fold cross-validation procedure, and validation ROC-AUC objective. Data preprocessing included K-Nearest Neighbors imputation, Min–Max normalization, and Synthetic Minority Oversampling Technique (SMOTE), with preprocessing parameters estimated from the training partition within each cross-validation fold. The SMO-optimized DNN achieved a mean accuracy of 0.6947 and a mean ROC-AUC of 0.6955 across the ten folds, followed by a held-out test ROC-AUC of 0.6640. Among the compared metaheuristic methods, SMO obtained the highest held-out test ROC-AUC among the four metaheuristic methods, whereas GWO achieved the highest F1-score, indicating that the relative performance of the optimization methods depended on the evaluation metric. Post hoc statistical analysis showed significant ROC-AUC differences between SMO and GWO and between SMO and FA, while the difference between SMO and PSO was not statistically significant. Permutation Feature Importance (PFI) identified Sulfate, pH, and Hardness as the highest-ranked features for the trained SMO-DNN according to their contribution to ROC-AUC. Using the six highest-ranked features increased recall, F1-score, and ROC-AUC relative to the full-feature model, although computation time was not reduced. Overall, the results indicate that SMO is a competitive alternative for DNN hyperparameter optimization in this water potability classification setting, particularly when ROC-AUC is used as the primary optimization objective. However, the moderate predictive performance and relatively low recall for the potable class indicate that further improvements are still required.

Keywords Spider Monkey Optimization, Deep Neural Network, Heuristic, Permutation Feature Importance, Water Potability

AMS 2010 subject classifications : 68T05, 68T20, 90C59

DOI: 10.19139/soic-2310-5070-3627

1. Introduction

Accurate prediction of water quality is essential for safeguarding public health and supporting sustainable environmental management [1, 2]. However, water-potability classification from physicochemical measurements remains challenging because the relationships among water-quality indicators and potability can be nonlinear, while practical datasets may also contain missing observations and unequal class distributions [3]. Deep neural networks (DNNs) provide a flexible approach for modeling nonlinear relationships among multiple predictors [4, 5]. Despite their advantages, DNNs are highly sensitive to the selection of hyperparameters and input features; suboptimal configurations often result in overfitting or poor generalization [6, 7].

*Correspondence to: Syaiful Anam (Email: syaiful@ub.ac.id). Department of Mathematics, Faculty of Mathematics and Natural Science, University of Brawijaya Malang, Indonesia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

To improve hyperparameter optimization, recent efforts have shifted from exhaustive search methods, such as grid search [8], toward metaheuristic-based approaches. Metaheuristic algorithms, including Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and the Firefly Algorithm (FA), have been widely applied in this context [9–11]. Nevertheless, limitations have been reported in their performance. PSO is prone to premature convergence in complex search landscapes [12]; GWO may suffer from reduced population diversity in later iterations [13]; and FA's efficiency is highly sensitive to parameter tuning [14]. Spider Monkey Optimization (SMO) provides a distinct population-based search mechanism inspired by fission–fusion social behavior, in which local and global leaders guide candidate solutions during the search process [15]. This structure provides a relevant basis for investigating SMO as a DNN hyperparameter optimization strategy, but does not imply that SMO should be expected to outperform other optimizers in every setting. Recent studies have demonstrated the effectiveness of SMO in optimizing neural network models for various environmental modeling tasks [16,17]. However, systematic comparisons between SMO and established metaheuristics, particularly in Receiver Operating Characteristic Area Under the Curve (ROC-AUC) oriented tasks involving imbalanced datasets, remain limited.

Model performance is also influenced by the information contributed by the input variables. Permutation Feature Importance (PFI) provides a model-agnostic post hoc approach for estimating predictive importance by measuring the change in model performance after the values of an individual feature are permuted [18, 19]. While SMO has been applied in related tasks [20], evidence remains limited on its combined use with permutation-based feature analysis for water-potability classification under a controlled comparison with established metaheuristic algorithms.

This study proposes a hybrid SMO–PFI–DNN framework to address this research gap. The primary objectives are: (1) to assess SMO as a DNN hyperparameter optimization strategy relative to PSO, GWO, and FA under a common hyperparameter search space, population size, number of iterations, and validation ROC-AUC objective; (2) to apply PFI to the SMO-optimized DNN to examine model-dependent feature relevance and to evaluate the predictive behavior of the model after reducing the input feature set; and (3) to place the optimized DNN results in a broader predictive context through comparison with non-DNN classifiers and complementary evaluation metrics. The study contributes the following: (i) a controlled comparison of SMO, PSO, GWO, and FA for DNN hyperparameter optimization under a common experimental setting; (ii) the integration of permutation-based feature analysis with the SMO-optimized DNN to investigate model-dependent feature relevance and the effect of reducing the input feature set; (iii) the inclusion of Random Forest (RF) and XGBoost as supplementary non-DNN predictive baselines to provide broader context for the performance of the optimized DNN; and (iv) an evaluation of the resulting models using multiple performance criteria and computational characteristics, enabling metric-dependent strengths and limitations to be discussed without assuming that a single method is uniformly superior.

The remainder of this paper is organized as follows. Section 2 reviews related studies on water-quality prediction, metaheuristic neural-network optimization, and feature analysis. Section 3 describes the dataset, preprocessing procedures, SMO–DNN integration, PFI procedure, and experimental setup. Section 4 presents and discusses the experimental results, including comparisons with other metaheuristics and non-DNN baselines, feature-importance analysis, and statistical evaluation. Finally, Section 5 summarizes the main findings, limitations, and directions for future research.

2. Related Work

Water-potability prediction has been investigated using a range of machine-learning and neural-network approaches. These studies generally aim to capture nonlinear relationships among physicochemical water-quality indicators while addressing practical issues such as missing observations and unequal class distributions. Within the SOIC literature, Nachaoui et al. [21] formulated water-safety prediction as a supervised-learning problem combined with regularized optimization. Their study demonstrated the applicability of optimization-based machine learning to water-quality classification; however, it did not address DNN hyperparameter optimization using population-based metaheuristics or post hoc feature-importance analysis.

Metaheuristic optimization has also been investigated for configuring and improving neural-network models across different application domains. Khare et al. [22] combined SMO with a DNN for intrusion detection, whereas Aarthi et al. [23] employed a balanced SMO variant with a Bi-LSTM architecture for air-quality prediction. Kumari et al. [16] further examined SMO-based neural-network optimization for intrusion detection. Collectively, these studies demonstrate that SMO can be incorporated into neural-network learning frameworks. Nevertheless, differences in application domain, network architecture, optimization objective, and evaluation protocol limit the extent to which their reported results can be generalized to water-potability classification.

The need for controlled comparisons among optimization methods is also supported by recent benchmarking research. Eddaoudi et al. [24] compared several metaheuristic approaches for neural-network optimization across multiple machine-learning datasets and reported different trade-offs between training performance, generalization, and execution time. Their findings indicate that the relative performance of a metaheuristic depends on both the problem setting and the evaluation criterion. This observation is relevant to the present study, where ROC-AUC is used as the primary optimization objective, while precision, recall, and F1-score are examined as complementary measures of classification behavior.

A related line of research concerns feature selection and feature-importance analysis. Sahu et al. [25] proposed a hybrid wrapper approach combining SMO and simulated annealing for feature selection. More recently, Altaee et al. [26] integrated metaheuristic hyperparameter optimization and feature selection within a CNN-based classification framework. These optimization-based feature-selection approaches differ conceptually from PFI. Whereas wrapper or metaheuristic feature-selection methods directly search for a subset of input variables, PFI is a post hoc procedure that evaluates how strongly the predictive performance of an already trained model depends on each input variable [18, 19]. This distinction is important because PFI scores describe model-dependent predictive importance rather than causal influence.

Taken together, previous studies provide evidence for three components relevant to the present work: machine-learning approaches for water-quality classification, metaheuristic optimization of neural networks, and feature-selection or feature-importance analysis. However, evidence remains limited on their combined evaluation within a controlled water-potability classification framework in which SMO, PSO, GWO, and FA optimize the same DNN hyperparameter space under a common population size, number of iterations, and validation ROC-AUC objective, followed by permutation-based analysis of the optimized model. The present study therefore focuses on comparing the search behavior and predictive outcomes of these metaheuristic optimizers under a common experimental setting, while RF and XGBoost are included only as supplementary non-DNN predictive baselines to provide broader context for the resulting DNN performance.

3. Methods

A structured experimental methodology was employed to evaluate the effects of hyperparameter optimization and feature analysis on the performance of a DNN for water-potability classification. The procedure comprised four main stages: (i) data preprocessing, (ii) DNN hyperparameter optimization using SMO, (iii) post hoc feature-importance analysis using PFI, followed by evaluation of a reduced input feature set, and (iv) model evaluation using cross-validation and a held-out test set. The primary optimization criterion was ROC-AUC. An overview of the experimental workflow is illustrated in Figure 1.

3.1. Dataset

The experiments were conducted using the publicly available Water Potability dataset, obtained from Kaggle (https://www.kaggle.com/datasets/adityakadiwal/water-potability). This dataset is widely used for developing machine learning and deep learning models aimed at classifying water safety based on various physical and chemical indicators. The dataset contains 3,276 samples, each comprising nine numerical features and a binary target variable labeled Potability, where 1 indicates potable (safe) water, and 0 indicates non-potable water. The input features include water quality indicators such as: pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, and Turbidity.

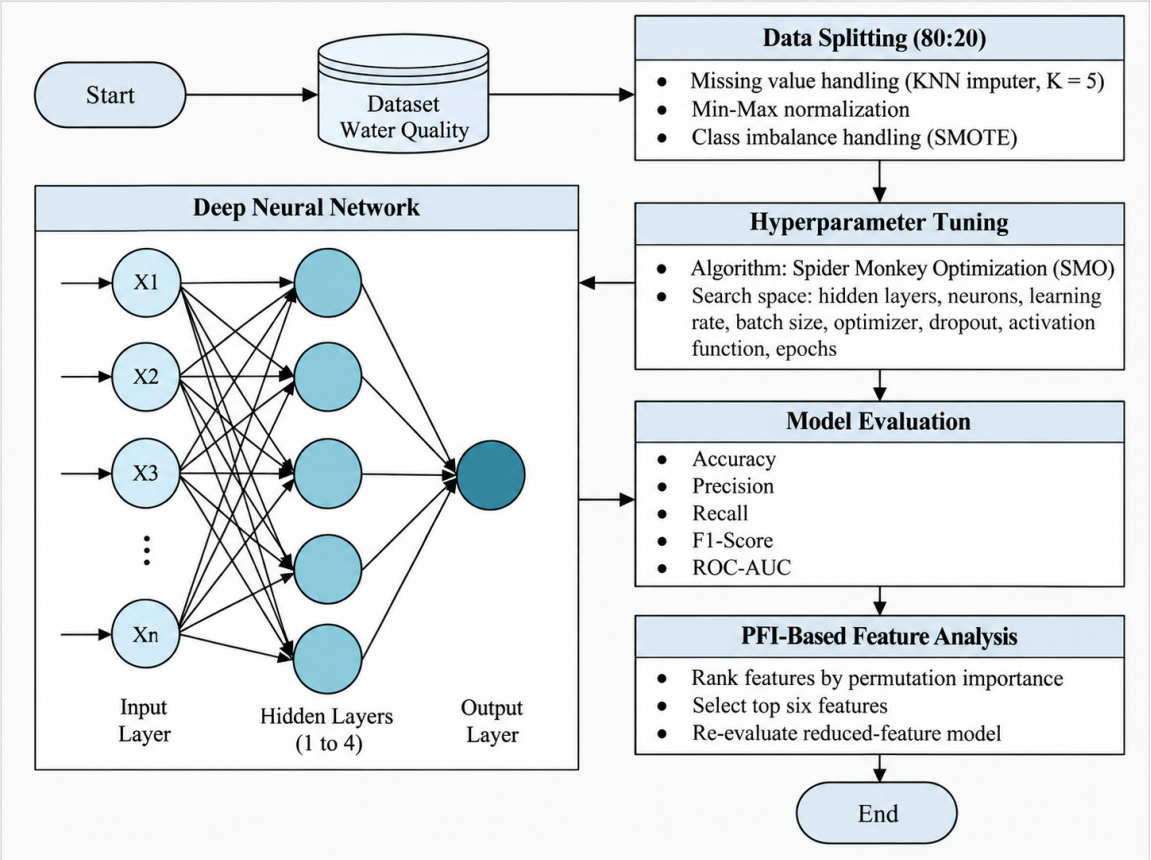


Figure 1. Workflow of the proposed SMO–PFI–DNN framework

Descriptive statistics for each feature are presented in Table 1. The features exhibit substantial differences in numerical scale; for example, the values of Solids are considerably larger than those of Turbidity. Missing observations are present in the pH, Sulfate, and Trihalomethanes features, indicating the need for an explicit imputation procedure before model training.

3.2. Data Preprocessing

Data preprocessing was performed to handle missing observations, place the numerical predictors on a comparable scale, and address class imbalance before model training. The preprocessing steps consisted of K-Nearest Neighbors (KNN) imputation, Min–Max normalization, and the Synthetic Minority Oversampling Technique (SMOTE). Within each cross-validation fold, the KNN imputer and Min–Max scaler were fitted exclusively on the training partition and subsequently applied to the corresponding validation partition. SMOTE was then applied only to the preprocessed training partition, while the validation data remained unchanged.

3.2.1. ***Handling Missing Values.*** Missing values were identified in the *pH*, *Sulfate*, and *Trihalomethanes* features. KNN imputation with $K = 5$ was used to estimate the missing values based on neighboring observations in the feature space [27]. Unlike univariate imputation based only on a feature mean or median, KNN imputation uses information from multiple predictors when estimating a missing entry.

Table 1. Statistical Description of Dataset Features

Feature	Description	Mean	Std	Min	Max	Missing
pH	pH level of the water (scale 0–14)	7.08	1.594	0.0	13.999	491
Hardness	Hardness of the water (mg/L)	196.369	32.879	47.432	323.124	0
Solids	Total dissolved solids (mg/L)	22014.092	8768.570	320.942	61227.196	0
Chloramines	Chloramines concentration (mg/L)	7.122	1.583	0.352	13.127	0
Sulfate	Sulfate concentration (mg/L)	333.775	41.416	129.000	481.030	781
Conductivity	Water conductivity (μS/cm)	426.205	80.824	181.483	753.342	0
Organic_carbon	Organic carbon concentration (mg/L)	14.284	3.308	2.199	28.300	0
Trihalomethanes	Trihalomethane concentration (μg/L)	66.396	16.175	0.737	124.000	162
Turbidity	Turbidity level (NTU)	3.966	0.780	1.450	6.739	0
Potability	Potability status (1 = potable, 0 = not)	0.390	0.487	0.0	1.0	0

3.2.2. ***Data Normalization.*** All numerical features were normalized using Min–Max scaling to transform their values into the 0–1 range. This step places predictors with different numerical ranges on a comparable scale prior to DNN training [28].

3.2.3. ***Addressing Class Imbalance.*** The dataset exhibits a class imbalance, with approximately 61% of samples labeled as non-potable and 39% as potable. Such an imbalance may cause the classifier to favor the majority class and can reduce sensitivity to the minority class. To mitigate this issue, SMOTE was applied. SMOTE generates synthetic minority-class observations by interpolating between neighboring minority samples [29]. SMOTE was applied only to the training partition within each cross-validation fold; no synthetic samples were introduced into the validation or held-out test data [30].

3.3. Spider Monkey Optimization (SMO)

SMO is a population-based metaheuristic inspired by the fission–fusion social structure of spider monkeys during foraging, as originally proposed by Bansal et al. [31]. Each spider monkey represents a candidate solution in a D-dimensional search space; in this study, a candidate solution corresponds to a DNN hyperparameter configuration. The search process is organized through local and global leadership mechanisms and consists of six main phases: Local Leader Phase (LLP), Global Leader Phase (GLP), Local Leader Learning Phase, Global Leader Learning Phase, Local Leader Decision Phase, and Global Leader Decision Phase.

Population Initialization. Each spider monkey's position is initialized randomly within a D-dimensional search space, as expressed in Equation (1):

$$SM_{ij} = SM_j^{\min} + U(0,1) \times \left(SM_j^{\max} - SM_j^{\min}\right) \tag{1}$$

where SM_{ij} is the j-th dimension of the i-th spider monkey's position vector, $U(0,1)$ is a random scalar sampled from a uniform distribution, and $SM_j^{\min}$ and $SM_j^{\max}$ denote the lower and upper bounds of the j-th hyperparameter.

Local Leader Phase (LLP). For each dimension, the position update is performed according to the perturbation rate pr. When the update condition is satisfied, the new position is generated as:

$$SM_{ij}^{new} = SM_{ij} + U(0,1)\,(LL_{kj} - SM_{ij}) + U(-1,1)\,(SM_{rj} - SM_{ij}) \tag{2}$$

where LL_{kj} denotes the j-th dimension of the local leader of group k, and SM_{rj} denotes the j-th dimension of a randomly selected member from the same group. If the perturbation condition is not satisfied, the corresponding dimension remains unchanged.

Global Leader Phase (GLP). The probability of selecting the i-th spider monkey for an update is computed as:

$$prob_i = 0.9 \cdot \frac{fit_i}{fit_{\max}} + 0.1 \tag{3}$$

if selected, the update rule is:

$$SM_{ij}^{new} = SM_{ij} + U(0,1)\,(GL_j - SM_{ij}) + U(-1,1)\,(SM_{rj} - SM_{ij}) \tag{4}$$

where GL_j denotes the position of the global leader.

Learning Phases. The local and global leaders update their positions only when superior solutions are found within their respective groups or the entire population. If no improvement is observed over a predefined number of iterations, the algorithm proceeds to the decision phases.

Local Leader Decision Phase. When a local leader does not improve for a predefined Local Leader Limit, the members of the corresponding group are redirected either through random reinitialization within the search bounds or through an update that combines information from the global and local leaders. For the latter case,

$$SM_{ij}^{new} = SM_{ij} + U(0,1)\,(GL_j - SM_{ij}) + U(0,1)\,(SM_{ij} - LL_{kj}) \tag{5}$$

Global Leader Decision Phase. When the global leader stagnates, the population undergoes fission (splitting into subgroups) or fusion (merging subgroups), thereby preserving diversity in the search space.

3.4. Deep Neural Network (DNN)

In this study, the DNN was used to model nonlinear relationships between the nine physicochemical predictors and the binary potability outcome. Rather than fixing a single network architecture in advance, several architectural and training hyperparameters were determined through the SMO-based optimization process.

The DNN consists of an input layer, one or more hidden layers, and a single output neuron. The number of hidden layers, number of neurons per layer, activation function, dropout rate, learning rate, batch size, and optimizer were included in the hyperparameter search space. The number of training epochs was not treated as an optimization variable; instead, a common maximum training limit of 100 epochs with early stopping was applied as described below.

Input Layer. The input layer receives nine preprocessed numerical features extracted from the Water Potability dataset. These features represent physical and chemical properties of the water samples and serve as the foundation for downstream classification.

Hidden Layers The number of hidden layers and the number of neurons per layer were varied during the hyperparameter optimization process. ReLU and Leaky ReLU were included as candidate activation functions. Dropout was applied after each hidden layer as a regularization mechanism, with the dropout rate included in the search space over the range 0.1–0.5 [32].

Output Layer The output layer consists of a single neuron activated via a sigmoid function, enabling the model to perform binary classification (i.e., potable vs. non-potable water). This configuration transforms the network's final output into a probability score between 0 and 1, indicating the likelihood that a given sample is potable.

Loss Function and Optimizer Binary cross-entropy was used as the loss function for the binary classification task [33]. The optimizer was treated as a tunable hyperparameter, with Adam, RMSProp, and Stochastic Gradient Descent (SGD) included as candidate optimization algorithms.

Regularization and Early Stopping Early stopping was applied uniformly to all candidate DNN models by monitoring validation ROC-AUC. Each model was allowed to train for a maximum of 100 epochs, while training was terminated earlier when validation ROC-AUC failed to improve for 10 consecutive epochs. The patience value of 10 epochs was used as a fixed stopping window to tolerate short-term fluctuations in validation performance while avoiding unnecessarily prolonged training after validation performance had plateaued. Because `restore_best_weights=True` was enabled, the model weights corresponding to the best observed validation ROC-AUC were restored before evaluation.

3.5. Integration of SMO and DNN

SMO was integrated with the DNN training and validation procedure to search for hyperparameter configurations that maximize validation ROC-AUC. Each spider monkey represents one candidate DNN configuration. For every candidate, the model was evaluated using 10-fold cross-validation, and the mean validation ROC-AUC across the ten folds was used as its fitness value.

The integration process consists of the following steps:

3.5.1. ***Population Initialization.*** An initial population of 50 spider monkeys was randomly generated within the predefined DNN hyperparameter search space. The optimized hyperparameters included the number of hidden layers, number of neurons per layer, learning rate, batch size, optimizer, dropout rate, and activation function.

3.5.2. ***Position Decoding to DNN Configuration.*** Each position vector in the SMO population was decoded into a valid DNN configuration. Continuous and discrete components of the position vector were mapped to their corresponding hyperparameter values according to the predefined search ranges.

3.5.3. ***DNN Model Training.*** For each decoded configuration, a DNN model was instantiated and evaluated through 10-fold cross-validation. Within each fold, the preprocessing pipeline described in Section 3.2 was applied exclusively using the corresponding training partition before model fitting. The number of training epochs was not included as an optimization variable in the metaheuristic search. Instead, all candidate configurations were trained with a common maximum limit of 100 epochs. Early stopping monitored validation ROC-AUC and terminated training when no improvement was observed for 10 consecutive epochs. This patience window was applied uniformly across all candidate configurations to allow short-term fluctuations in validation performance while limiting unnecessarily prolonged training. The patience setting therefore represents a predefined stopping trade-off rather than a guarantee that no later improvement could theoretically occur. The weights corresponding to the best observed validation ROC-AUC were restored before model evaluation.

3.5.4. ***Performance Evaluation (Fitness Calculation).*** For each candidate configuration, ROC-AUC was calculated on the validation partition of each of the ten folds. The mean validation ROC-AUC across the 10-fold cross-validation procedure was then used as the fitness value of the corresponding spider monkey. ROC-AUC was selected as the optimization objective because it evaluates the ranking ability of the classifier across classification thresholds and is less dependent on a single decision threshold than threshold-based measures such as accuracy or F1-score.

3.5.5. ***Position Update via SMO.*** After fitness evaluation, the spider monkey positions were updated according to the six phases of SMO described in Section 3.3. Candidate positions generated by these phases were decoded and evaluated using the same DNN training and cross-validation procedure.

3.5.6. ***Stopping Criterion.*** The optimization process was performed using a population size of 50 and was run for a fixed 100 iterations. The same population size and number of iterations were used for PSO, GWO, and FA to maintain a common optimization setting across the four metaheuristic methods.

3.5.7. ***Final Model Training.*** After completion of the optimization process, the hyperparameter configuration with the highest mean validation ROC-AUC was selected. Using this configuration, the final DNN model was trained on the development data according to the preprocessing and training procedures described above and was subsequently evaluated on the held-out test set. The held-out test data were not involved in hyperparameter optimization or candidate selection. The same maximum training limit of 100 epochs and early-stopping configuration described in Section 3.5.3 were retained during final-model training.

As illustrated in Figure 2, each optimization cycle involves the decoding of spider monkey positions into DNN configurations, model training, fitness evaluation, and position updating.

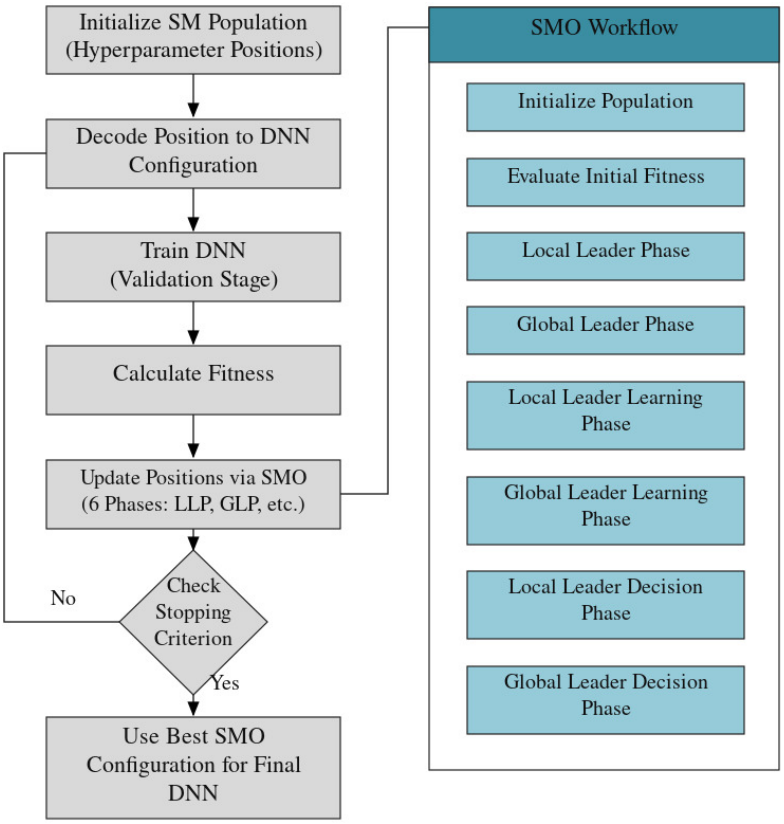


Figure 2. Integration of SMO and DNN

3.6. Permutation Feature Importance (PFI)

PFI was used as a post hoc, model-agnostic procedure to assess the predictive importance of the input features for the trained DNN. PFI measures the change in model performance after the values of one feature are randomly

permuted while the remaining features are kept unchanged. A larger decrease in performance indicates greater dependence of the model's predictions on the permuted feature [18, 19].

ROC-AUC was used as the performance metric for computing permutation importance. For each feature, the permutation procedure was repeated $K = 10$ times, and the mean decrease in ROC-AUC relative to the baseline score was used as the importance value. Features were ranked according to their mean importance scores, and the six highest-ranked features were retained for the reduced-feature experiment to examine the effect of removing the three lower-ranked inputs on predictive performance.

The resulting PFI values were interpreted as model- and metric-dependent measures of predictive importance. Accordingly, they indicate the extent to which the trained DNN relies on a feature for its ROC-AUC performance and should not be interpreted as evidence of causal or intrinsic importance in water quality.

The implementation steps are described in Algorithm 1.

Algorithm 1 Permutation Feature Importance Procedure

Inputs:
Trained DNN model: m
Validation dataset: D_{val}
Number of permutation repetitions: $K = 10$

Output:
Importance score i_j for each feature f_j

Compute the baseline ROC-AUC:
$s_{base} \leftarrow \text{ROC-AUC}(m, D_{val})$
for each feature $j = 1, 2, \ldots, d$ **do**
    for each repetition $k = 1, 2, \ldots, K$ **do**
        Create $\widetilde{D}_{k,j}$ by randomly permuting the j-th feature in D_{val}
        Compute $s_{k,j} \leftarrow \text{ROC-AUC}(m, \widetilde{D}_{k,j})$
    end for
    Compute the importance score:
    $$i_j = s_{base} - \frac{1}{K} \sum_{k=1}^{K} s_{k,j}$$
end for
Rank the features in descending order of i_j

Here, s_{base} denotes the baseline validation ROC-AUC, $s_{k,j}$ denotes the ROC-AUC after feature j is permuted in repetition k, and i_j denotes the resulting importance score for feature j.

3.7. Experimental Setup

SMO, PSO, GWO, and FA were applied independently to the same DNN hyperparameter optimization problem. The four methods used the same DNN hyperparameter search space, population size, number of iterations, cross-validation procedure, and validation ROC-AUC objective to provide a controlled comparison of their search performance.

3.7.1. Algorithm Configuration Each optimizer used a population size of 50 and was run for a fixed 100 iterations. The algorithm-specific control parameters used in the experiments are summarized in Table 2, together with the corresponding methodological references.

Table 2. Configuration of Compared Algorithms

Algorithm	Specific Parameters	Reference
PSO	Inertia = 0.7, c_1 = 1.5, c_2 = 1.5	Kennedy & Eberhart, 1995 [34]
GWO	Standard position update using α, β, and δ leaders	Mirjalili et al., 2014 [35]
FA	β_0 = 1, γ = 1, α = 0.2	Yang, 2010 [36]
SMO	MG = $N/4$, GLL = N, LLL = DN, pr = 0.3	Bansal et al., 2014 [31]

3.7.2. Hyperparameter Search Space All four algorithms searched over the same DNN hyperparameter space. The search space was constructed to balance model complexity, training stability, and generalization ability. The selected hyperparameters, their respective ranges, and justification for inclusion are summarized in Table 3.

Table 3. Common DNN Hyperparameter Search Space

Hyperparameter	Range	Justification
Hidden Layer	1–4 layers	Determines the complexity of the network's representation. Too few layers may lead to underfitting, while too many increase the risk of overfitting [33].
Neuron/Units	16–256 neurons	Controls the processing capacity of each layer. Higher dimensions allow the network to represent more complex feature interactions.
Learning Rate	0.0001–0.1	Influences the convergence process. A rate that is too high may cause instability, whereas a rate that is too low may slow convergence [37].
Batch Size	16, 32, 64, 128	Controls the number of samples used in each gradient update and may affect both training behavior and generalization.
Optimizer	Adam, RMSProp, SGD	Provides alternative gradient-based optimization strategies for DNN training [38].
Dropout Rate	0.1–0.5	Controls the level of dropout regularization applied during training.
Activation Function	ReLU, LeakyReLU	Defines the nonlinear transformation used in the hidden layers.

3.7.3. Training and Evaluation Environment The four metaheuristic methods were evaluated using the common DNN training and 10-fold cross-validation procedure described in Section 3.5, with validation ROC-AUC used as the optimization objective.

All experiments were implemented in Python 3.9 using TensorFlow 2.12 for DNN modeling and Scikit-Learn 1.2 for preprocessing, cross-validation, and performance evaluation. The training and optimization procedures were executed on the Nebula High-Performance Computing (HPC) cluster using the PBS job scheduler. Each job used a single compute node with 64 CPU cores and 32 GB RAM. GPU acceleration was not used, so the compared metaheuristic experiments were conducted under the same CPU-based computational setting.

3.7.4. Evaluation Metrics Model performance was assessed using five primary classification metrics: accuracy, precision, recall, F1-score, and ROC-AUC. The positive class was defined as *Potability*=1. In addition, the held-out test predictions were examined using a confusion matrix to characterize class-specific prediction errors.

- **Accuracy**: the proportion of correctly classified samples.
- **Precision**: the proportion of predicted potable samples that were correctly classified.
- **Recall (Sensitivity)**: the proportion of actual potable samples that were correctly identified.

- **F1-Score**: the harmonic mean of precision and recall at the selected classification threshold.
- **ROC-AUC**: the area under the Receiver Operating Characteristic curve, measuring discrimination across classification thresholds.

For threshold-dependent measures, a classification threshold of 0.5 was used. During hyperparameter optimization, the mean validation ROC-AUC across the ten folds was used as the fitness value, while accuracy, precision, recall, and F1-score were recorded as complementary performance measures. After hyperparameter selection, the final model was evaluated on the held-out test set using the same five metrics, with the confusion matrix used as an additional error analysis.

3.7.5. Supplementary Non-DNN Predictive Baselines To provide additional context for the predictive performance of the SMO-optimized DNN, RF [39] and XGBoost [40] were included as supplementary non-DNN classification baselines. These models were not treated as competitors in the metaheuristic search comparison because they do not optimize DNN hyperparameters. Instead, they were evaluated as alternative predictive models for the same water-potability classification task.

Both baseline models were evaluated using a stratified 80:20 development–test split and 10-fold stratified cross-validation on the development data. The preprocessing procedure consisted of KNN imputation, Min–Max scaling, and SMOTE, with all preprocessing steps fitted only on the training partition within each cross-validation fold. Hyperparameter selection was performed using randomized search over 40 candidate configurations for each model, with mean validation ROC-AUC used as the selection criterion. The selected model was then evaluated once on the held-out test set. Because RF and XGBoost were included as supplementary predictive baselines rather than metaheuristic optimization methods, their hyperparameter-search budget and search runtime were not directly compared with the iteration-based computational cost of SMO, PSO, GWO, and FA.

4. Results and Discussion

This section presents the predictive performance of the SMO-optimized DNN, its comparison with the other metaheuristic methods, the PFI-based feature analysis, and the statistical comparison of the optimization methods. Table 4 summarizes the 10-fold cross-validation results of the SMO-optimized DNN on the development data. The model achieved a mean accuracy of 0.6947 and a mean ROC-AUC of 0.6955. Across the ten folds, ROC-AUC values ranged from 0.6599 to 0.7363, indicating some variation in predictive discrimination across the validation folds.

The mean precision was 0.6768, whereas the mean recall for the potable class was lower at 0.4374, resulting in a mean F1-score of 0.5236. This difference indicates that, at the classification threshold used for the threshold-dependent metrics, the model identified only a portion of the potable samples. Because ROC-AUC was the primary optimization objective, the comparatively lower recall and F1-score also illustrate that performance conclusions depend on the evaluation metric being considered.

Figure 3 illustrates the training and validation behavior of the final SMO-optimized DNN in terms of accuracy and loss. These curves are presented as diagnostic indicators of training dynamics rather than as the criterion used for early stopping, which was based on validation ROC-AUC as described in Section 3.5.3. The training loss decreased progressively throughout training, whereas the validation loss initially decreased but subsequently fluctuated and increased more markedly during the later training stage. The growing divergence between the training and validation loss indicates that continued improvement on the training data did not translate into a corresponding improvement in validation behavior and is consistent with late-stage overfitting. Accordingly, the decreasing training loss alone should not be interpreted as evidence of uniformly stable convergence.

Early stopping was used as a training safeguard by monitoring validation ROC-AUC with a patience of 10 consecutive epochs without improvement. Because `restore_best_weights=True` was enabled, the weights corresponding to the best observed validation ROC-AUC were restored before model evaluation rather than retaining the weights from the final stopping epoch. Figure 3 displays accuracy and loss as diagnostic indicators

Table 4. 10-Fold Cross-Validation Results of SMO on Development Data

Fold	Accuracy	Precision	Recall	F1-Score	ROC-AUC
1	0.7214	0.8372	0.3529	0.4966	0.7148
2	0.6832	0.7209	0.3039	0.4276	0.7177
3	0.6985	0.6139	0.6078	0.6108	0.7363
4	0.6756	0.6164	0.4412	0.5143	0.6645
5	0.7023	0.6667	0.4706	0.5517	0.6775
6	0.6908	0.6842	0.3824	0.4906	0.6599
7	0.7214	0.6883	0.5196	0.5922	0.7211
8	0.6756	0.6164	0.4412	0.5143	0.6817
9	0.6832	0.6471	0.4272	0.5146	0.6790
10	0.6947	0.6769	0.4272	0.5238	0.7024
Avg	**0.6947**	**0.6768**	**0.4374**	**0.5236**	**0.6955**

of training behavior; therefore, the curves are not used to retrospectively identify the best validation ROC-AUC epoch.

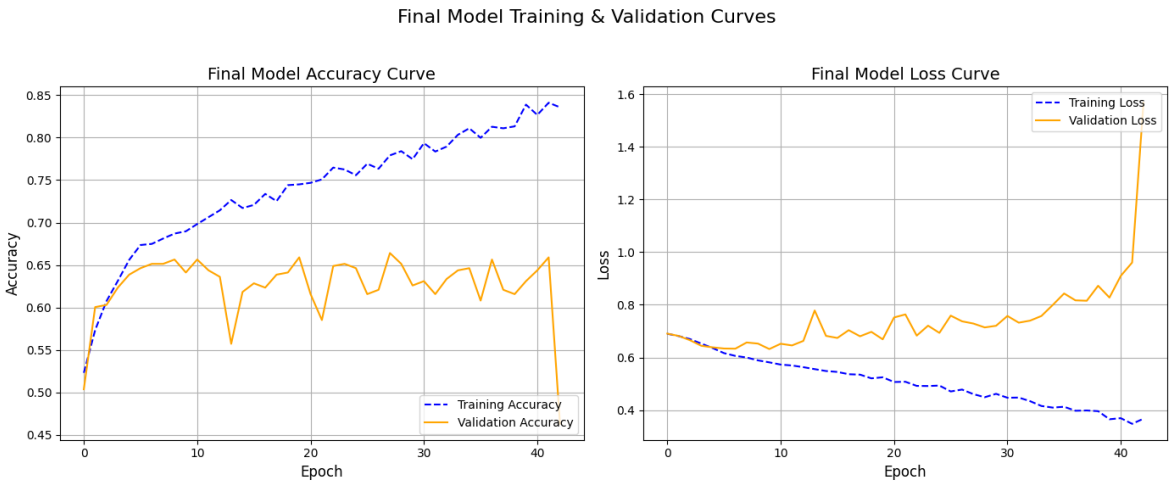


Figure 3. Training and validation accuracy and loss curves of the final SMO-optimized DNN.

Table 5. Best DNN Hyperparameters from SMO Optimization

Hyperparameter	Optimal Value
Hidden Layer	3
Neurons per Layer	128
Learning Rate	0.0943
Batch Size	32
Optimizer	SGD
Dropout Rate	0.4932
Activation Function	ReLU

The selected learning rate of 0.0943 lies close to the upper bound of the predefined search interval (0.0001–0.1). Importantly, this value was not selected independently; it emerged from the joint SMO-based optimization of the

learning rate together with network depth, number of neurons, batch size, optimizer, dropout rate, and activation function, using mean validation ROC-AUC as the fitness criterion. The value should therefore be interpreted in the context of the complete selected configuration, which combined a learning rate of 0.0943 with SGD, three hidden layers of 128 neurons, a batch size of 32, a dropout rate of 0.4932, and ReLU activation.

Because the selected learning rate is close to the upper boundary of the search space, it is reported as a configuration- and data-specific empirical outcome of the HPO procedure rather than as a generally recommended learning rate for DNN training. Furthermore, the training curves in Figure 3 do not support interpreting the decreasing training loss as evidence of uniformly stable convergence, since validation loss diverged during the later training stage. Training was therefore governed by a common maximum limit of 100 epochs and early stopping based on validation ROC-AUC with a patience of 10 epochs to limit prolonged training after validation performance ceased to improve. Because `restore_best_weights=True` was enabled, the weights corresponding to the best observed validation ROC-AUC were restored before evaluation. This stopping rule should be interpreted as a predefined training-control mechanism rather than as a guarantee that no later improvement could theoretically occur.

Using this configuration, the final model achieved an accuracy of 0.6616 and a ROC-AUC of 0.6640 on the held-out test set. The complete held-out test results are summarized in Table 6. The test ROC-AUC was lower than the mean cross-validation ROC-AUC of 0.6955, indicating a modest reduction in discriminative performance on unseen data. Recall for the potable class was 0.3750, with a corresponding F1-score of 0.4638, indicating that detection of potable samples remained a limitation of the model.

Table 6. Held-Out Test Performance of the SMO-Optimized DNN

Evaluation Metric	**Value**
Accuracy	0.6616
Precision	0.6076
Recall	0.3750
F1-Score	0.4638
ROC-AUC	0.6640

To further examine the classification errors on the held-out test set, a confusion-matrix analysis was conducted. The corresponding confusion matrix is presented in Table 7.

Table 7. Confusion Matrix of the SMO-Optimized DNN

	Predicted Non-Potable	**Predicted Potable**
Actual Non-Potable	338	62
Actual Potable	160	96

The misclassification pattern was strongly asymmetric between the two classes. Among the 256 potable samples in the held-out test set, 160 were incorrectly classified as non-potable and only 96 were correctly identified, corresponding to a false-negative rate of 62.5% and a sensitivity of 37.5%. In contrast, among the 400 non-potable samples, 338 were correctly classified and 62 were misclassified as potable, corresponding to a false-positive rate of 15.5% and a specificity of 84.5%. Of the 222 total misclassified samples, false negatives accounted for 72.1%. These results show that the dominant source of classification error at the 0.5 decision threshold was the failure to identify potable samples rather than the incorrect classification of non-potable samples.

This asymmetric error pattern is consistent with the relatively low recall and F1-score of the SMO-DNN. ROC-AUC evaluates discriminative ability across classification thresholds, whereas recall, F1-score, and the confusion matrix reflect model behavior at the fixed decision threshold of 0.5. Consequently, a model configuration with a comparatively favorable ROC-AUC does not necessarily achieve high sensitivity or F1-score at this threshold. The high false-negative rate therefore highlights a limitation in detecting potable samples at the selected operating point. This finding motivates consideration of alternative decision thresholds or cost-sensitive learning strategies in future work when improved sensitivity to the potable class is required.

4.1. Comparison with Other Metaheuristic Methods

The performance of SMO was compared with PSO, GWO, and FA under the common experimental setting described in Section 3.7. Figure 4 summarizes the predictive performance of the four methods across the evaluated metrics, while Figure 5 presents their computation times. The results indicate that no single method achieved the best performance across all evaluation metrics.

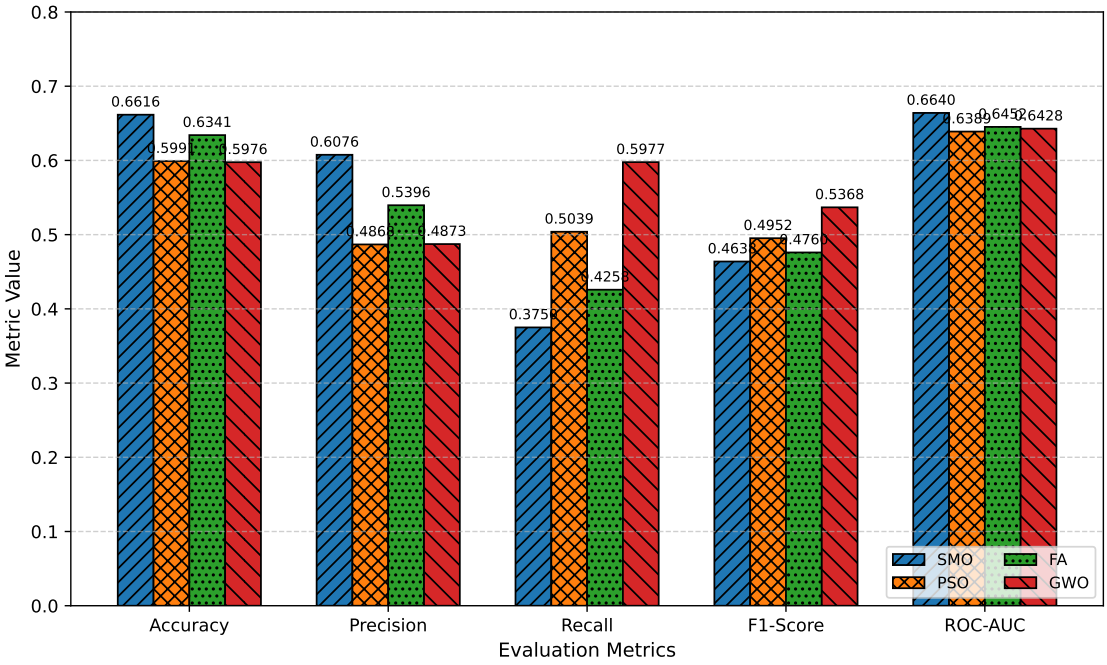


Figure 4. Performance Comparison Across Five Evaluation Metrics

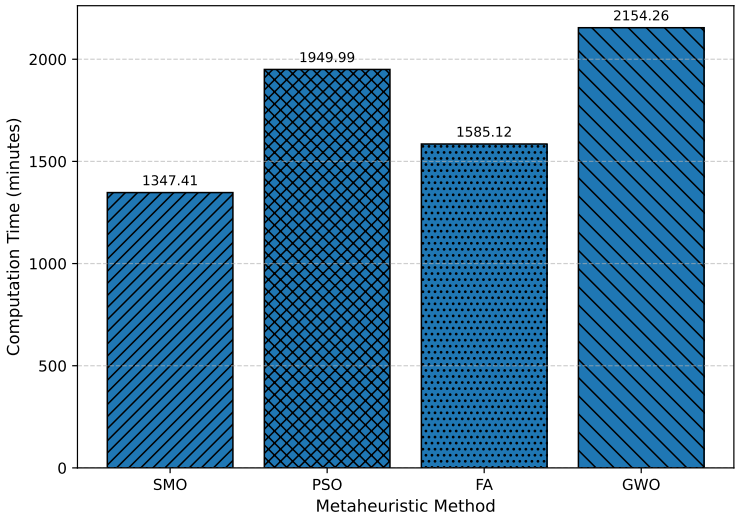


Figure 5. Computation Time of the Compared Metaheuristic Methods

GWO achieved the highest F1-score of 0.5368, whereas PSO and FA also obtained higher F1-scores than SMO. In contrast, SMO obtained the highest held-out test ROC-AUC of 0.6640 among the four metaheuristic methods. These results illustrate a metric-dependent trade-off. Mean validation ROC-AUC was prespecified as the fitness criterion during hyperparameter optimization, while the held-out test ROC-AUC was used to assess the generalization of the selected configuration. Accuracy, precision, recall, and F1-score were considered complementary measures of classification behavior.

The four methods also differed in computation time. SMO required 1347.41 minutes, whereas GWO required more than 2100 minutes under the same experimental setting. Thus, SMO required less computation time than GWO in the present experiment. However, this difference is treated as an empirical result of the implemented procedures and should not be interpreted as evidence that SMO is inherently more computationally efficient than the other metaheuristic methods.

Overall, the comparison does not support uniform superiority of one metaheuristic across all evaluation criteria. Moreover, as discussed in Section 4.4, the difference in cross-validation ROC-AUC between SMO and PSO was not statistically significant. The relative performance of the methods should therefore be interpreted according to the evaluation objective and the metric of interest rather than as a universal ranking of the optimizers.

4.2. Comparison with Non-DNN Predictive Baselines

To place the performance of the optimized DNN in a broader predictive context, the SMO-DNN was compared with RF and XGBoost as supplementary non-DNN baselines. Table 8 summarizes their cross-validation and held-out test performance.

Table 8. Predictive Performance of SMO-DNN and Non-DNN Baselines

Model	**CV ROC-AUC**	**Held-Out Test Performance**				
		Accuracy	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
SMO-DNN	0.6955	0.6616	0.6076	0.3750	0.4638	0.6640
RF	0.7001	0.6387	0.5401	0.5000	0.5193	0.6708
XGBoost	0.6930	0.6387	0.5375	0.5313	0.5344	0.6619

RF achieved the highest held-out test ROC-AUC among the three predictive models, with a value of 0.6708, compared with 0.6640 for SMO-DNN and 0.6619 for XGBoost. The three test ROC-AUC values were numerically close, with RF showing the highest value in this comparison. Thus, the SMO-optimized DNN did not demonstrate a clear predictive advantage over the two tree-based baselines.

The models also showed different behavior on the threshold-dependent metrics. SMO-DNN achieved higher test accuracy and precision, whereas RF and XGBoost achieved higher recall and F1-scores at the 0.5 classification threshold. In particular, recall increased from 0.3750 for SMO-DNN to 0.5000 for RF and 0.5313 for XGBoost. These results further illustrate the metric-dependent nature of the comparison: the relative performance of the models changes according to whether the emphasis is placed on overall discrimination, precision, or detection of the potable class.

4.3. Feature Importance Analysis

Figure 6 presents the PFI ranking obtained from the SMO-optimized DNN using ROC-AUC as the evaluation metric. Sulfate, pH, and Hardness showed the highest importance scores, indicating that the predictive performance of the trained model was more sensitive to permutations of these features than to permutations of the other inputs.

The PFI results should be interpreted as model-dependent predictive importance rather than as evidence of causal or intrinsic importance of the corresponding water-quality variables. In particular, a high importance score indicates that disrupting the information contained in a feature produced a larger reduction in ROC-AUC for the trained DNN. Conversely, low or negative importance values, as observed for some features, indicate that their permutation did not consistently reduce the model's ROC-AUC. Such values may reflect limited model dependence, redundancy with other predictors, interactions among features, or variation in the permutation procedure.

The comparatively lower PFI score of Solids should not be interpreted as evidence that solids are unimportant for water quality. Rather, for the trained SMO-DNN and the present dataset, permuting Solids produced a smaller decrease in ROC-AUC than permuting Sulfate or pH. This pattern may reflect limited model dependence, redundancy, or interactions with other predictors. Therefore, the observed PFI score represents model- and metric-dependent predictive importance rather than domain-level or causal importance.

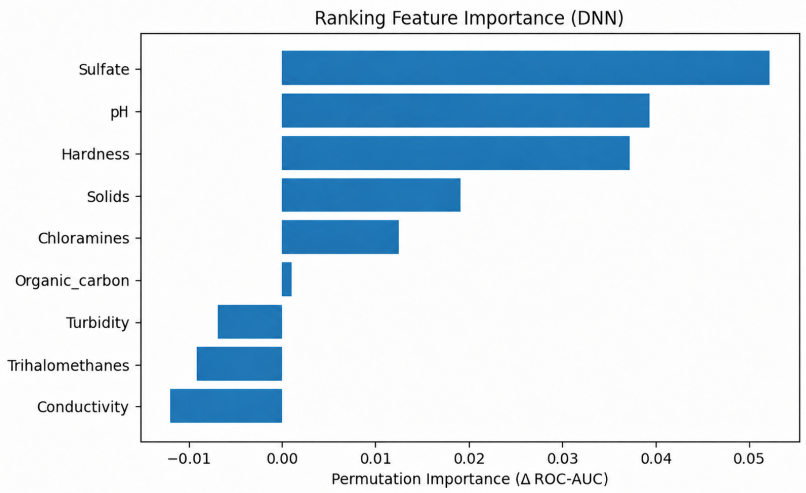


Figure 6. Feature Importance Ranking

From a water-quality perspective, the relatively high PFI rankings of Sulfate, pH, and Hardness are qualitatively consistent with the recognized relevance of these physicochemical characteristics in drinking-water assessment. The World Health Organization (WHO) considers pH, sulfate, hardness, and total dissolved solids among the physicochemical characteristics relevant to drinking-water quality, including operational and acceptability considerations [41]. In this context, the prominence of Sulfate, pH, and Hardness in the present PFI analysis is plausible from a domain perspective. However, the lower PFI of Solids, which represents total dissolved solids in this dataset, should not be interpreted as indicating lower intrinsic importance for water quality. Rather, it indicates that the ROC-AUC of the trained SMO-DNN was less sensitive to perturbations of Solids than to perturbations of Sulfate, pH, and Hardness under the present dataset and model configuration. Thus, the comparison with domain knowledge provides context for the observed PFI pattern without implying a causal or universal ranking of water-quality variables.

Based on the PFI ranking, the six highest-ranked features were retained for a reduced-feature experiment. The SMO-DNN was then evaluated using this reduced input set, and its predictive performance was compared with that obtained using all nine features. The results are summarized in Table 9.

Table 9. Performance Comparison: Full vs. Reduced Feature Set

Model	**Feature Count**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**	**Computation Time (min)**
SMO (all features)	9	0.6616	0.6076	0.3750	0.4638	0.6640	1347.41
SMO (selected features)	6	0.6738	0.6000	0.4922	0.5408	0.7117	1488.53

Using the six selected features, accuracy increased from 0.6616 to 0.6738, recall from 0.3750 to 0.4922, F1-score from 0.4638 to 0.5408, and ROC-AUC from 0.6640 to 0.7117, while precision decreased slightly from 0.6076 to 0.6000. These results indicate that, under the reported experimental setting, the six-feature model achieved better performance on four of the five reported classification metrics than the full-feature model.

However, reducing the input set from nine to six features did not result in a shorter computation time. The reduced-feature experiment required 1488.53 minutes, compared with 1347.41 minutes for the full-feature

experiment. A comparison of the optimized configurations provides additional empirical context for this result. Both models retained the same network depth and width, consisting of three hidden layers with 128 neurons per layer, and both used a batch size of 32 and ReLU activation. However, the reduced-feature model selected RMSprop with a learning rate of 0.0247 and a dropout rate of 0.0540, whereas the full-feature model selected SGD with a learning rate of 0.0943 and a dropout rate of 0.4932. Thus, the longer computation time of the reduced-feature experiment cannot be explained by an increase in network depth or width.

The observed runtime difference indicates that computational cost in the present HPO framework was not determined solely by the number of input features. Reducing the feature set changed the input representation presented to the optimization procedure, after which a different optimization configuration was selected. These differences in the selected optimization configurations may have been associated with different training trajectories and computational requirements during HPO. However, because the experiment was not designed as a controlled runtime ablation and the actual stopping epoch was not retained as a separate numerical output, the contribution of individual factors cannot be isolated conclusively. Therefore, the runtime difference is reported as an empirical outcome of the complete optimization pipeline and is not interpreted as evidence that feature reduction necessarily improves or degrades computational efficiency.

4.4. Statistical Significance Testing

To assess whether cross-validation ROC-AUC differed among SMO, PSO, GWO, and FA, a non-parametric Friedman test was applied using the ROC-AUC values from the corresponding folds of the 10-fold cross-validation procedure as paired observations. When the Friedman test indicated an overall significant difference, Nemenyi post hoc comparisons were used to examine pairwise differences among the optimization methods.

The Friedman test indicated a statistically significant overall difference among the four methods ($p < 0.05$), suggesting that at least one method differed from the others in terms of cross-validation ROC-AUC. The Nemenyi post hoc comparisons involving SMO are presented in Table 10. The differences between SMO and GWO ($p = 0.012$) and between SMO and FA ($p = 0.017$) were statistically significant at the 5% level, whereas the difference between SMO and PSO was not statistically significant ($p = 0.089$).

Table 10. Nemenyi Post Hoc Comparisons of SMO Against the Other Metaheuristic Methods

Comparison	p**-value**	**Statistically Significant**
SMO vs GWO	0.012	Yes
SMO vs FA	0.017	Yes
SMO vs PSO	0.089	No

These results indicate that the higher cross-validation ROC-AUC observed for SMO relative to GWO and FA was accompanied by statistically significant pairwise differences. In contrast, although SMO obtained a numerically higher cross-validation ROC-AUC than PSO, the difference was not statistically significant at the 5% level. Therefore, the results do not provide sufficient statistical evidence to conclude that SMO outperformed PSO in terms of ROC-AUC. The statistical findings should consequently be interpreted together with the observed performance metrics rather than as evidence of uniform superiority of SMO over all competing methods.

4.5. Limitations and Future Work

Several limitations should be considered when interpreting the results of this study. First, although SMOTE was applied to the training data to address class imbalance, recall for the potable class remained relatively low. Thus, the oversampling procedure did not eliminate the difficulty of identifying potable samples at the selected classification threshold. Future studies may investigate cost-sensitive loss functions, class weighting, focal loss, or decision-threshold adjustment to improve sensitivity to the potable class.

Second, metaheuristic hyperparameter optimization required substantial computation because each candidate configuration involved repeated DNN training within cross-validation. Future work may explore surrogate-assisted

optimization, multi-fidelity strategies, or other mechanisms for reducing the computational cost of the search process.

Third, ROC-AUC was used as the single optimization objective in this study. Consequently, the optimization process did not explicitly account for threshold-dependent measures such as recall and F1-score or for computational cost. Future research may therefore consider multi-objective formulations that jointly account for discriminative performance, minority-class detection, and computational requirements.

Fourth, the supplementary comparison with RF and XGBoost showed that the additional complexity of the SMO-DNN framework did not result in a clear predictive advantage over the tree-based baselines on this dataset. Future studies may investigate whether the relative benefit of metaheuristic-based DNN optimization becomes more apparent on larger datasets, higher-dimensional problems, or classification tasks with more complex nonlinear relationships.

Finally, the experiments were conducted using a single water-potability dataset. Evaluation on additional datasets would be useful for assessing the consistency and generalizability of the observed performance patterns across different water-quality data characteristics.

5. Conclusion

This study evaluated SMO for DNN hyperparameter optimization in water potability classification and compared its performance with PSO, GWO, and FA under a common hyperparameter search space, cross-validation procedure, and ROC-AUC-based optimization objective. Among the four metaheuristic methods, SMO achieved the highest held-out test ROC-AUC, whereas GWO achieved the highest F1-score. The statistical analysis of the cross-validation ROC-AUC further showed significant differences between SMO and GWO and between SMO and FA, while the difference between SMO and PSO was not statistically significant. These findings indicate that the relative performance of the optimization methods is metric-dependent and do not support a claim of uniform superiority of SMO across all evaluation criteria.

The supplementary comparison with RF and XGBoost provided additional context for the predictive performance of the optimized DNN. RF achieved the highest held-out test ROC-AUC among the three predictive models, while the SMO-DNN achieved higher accuracy and precision but lower recall and F1-score than the two tree-based baselines. These results indicate that the additional complexity of the SMO-DNN framework did not translate into a clear predictive advantage on this dataset.

PFI analysis identified Sulfate, pH, and Hardness as the highest-ranked features for the trained SMO-DNN according to their contribution to ROC-AUC. Using the six highest-ranked features increased accuracy, recall, F1-score, and ROC-AUC relative to the full-feature model in the reported experiment, although precision decreased slightly and computation time was not reduced. The findings support the use of PFI as a complementary model-interpretation and feature-ranking procedure, while its importance scores should be understood as model- and metric-dependent rather than as evidence of intrinsic or causal feature importance.

Overall, SMO provided a viable strategy for DNN hyperparameter optimization in this water potability classification setting, particularly when ROC-AUC was used as the primary optimization objective. However, the moderate predictive performance, relatively low recall for the potable class, and lack of a clear predictive advantage over the non-DNN baselines indicate that further improvement is required. Future work may investigate cost-sensitive learning, decision-threshold adjustment, multi-objective optimization, more computationally efficient hyperparameter-search strategies, and evaluation across additional water-quality datasets.

Acknowledgments

This research was funded by the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia and the Indonesian Education Foundation (LPDP) through the Center for Higher Education Funding and Assessment (PPAPT) under the Indonesian Education Scholarship (BPI) program.

REFERENCES

1. J. Zhou, J. Wang, Y. Chen, X. Li, and Y. Xie, *Water quality prediction method based on multi-source transfer learning for water environmental IoT system*, Sensors, vol. 21, no. 21, 7271, 2021.
2. Y. Chen, L. Song, Y. Liu, L. Yang, and D. Li, *A review of the artificial neural network models for water quality prediction*, Applied Sciences, vol. 10, no. 17, 5776, 2020.
3. W. Li, Y. Zhao, Y. Zhu, Z. Dong, F. Wang, and F. Huang, *Research progress in water quality prediction based on deep learning technology: a review*, Environmental Science and Pollution Research, vol. 31, no. 18, pp. 26415–26431, 2024.
4. A. Ferlitsch, *Deep Learning Patterns and Practices*, Manning, 2021.
5. M. El-Shebli, Y. Sharrab, and D. Al-Fraihat, *Prediction and Modeling of Water Quality Using Deep Neural Networks*, Environ. Dev. Sustain., vol. 26, no. 5, pp. 11397–11430, 2023.
6. J. A. Ilemobayo, O. Durodola, O. Alade, O. J. Awotunde, A. T. Olanrewaju, O. Falana, A. Ogungbire, A. Osinuga, D. Ogunbiyi, A. Ifeanyi, I. E. Odezuligbo, and O. E. Edu, *Hyperparameter tuning in machine learning: A comprehensive review*, Journal of Engineering Research and Reports, vol. 26, no. 6, pp. 388–395, 2024.
7. L. Owen, *Hyperparameter Tuning with Python: Boost your machine learning model's performance via hyperparameter tuning*, Packt Publishing, 2022.
8. M. R. Hossain and D. Timmer, *Machine learning model optimization with hyper parameter tuning approach*, Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence, vol. 21, no. 2, p. 31, 2021.
9. S. Kılıçarslan, *PSO+GWO: a hybrid particle swarm optimization and Grey Wolf optimization based algorithm for fine-tuning hyper-parameters of convolutional neural networks for cardiovascular disease detection*, Journal of Ambient Intelligence and Humanized Computing, vol. 14, pp. 87–97, 2022.
10. H. Fakhouri, F. Hamad, and A. K. Al Hwaitat, *SVM hyperparameter tuning using Grey Wolf Optimizer for heart disease prediction*, in Proc. 2025 1st International Conference on Computational Intelligence Approaches and Applications (ICCIAA), pp. 1–8, 2025.
11. L. F. Ahyar, S. Suyanto, and A. Arifianto, *Firefly algorithm-based hyperparameters setting of DRNN for weather prediction*, in Proc. 2020 International Conference on Data Science and Its Applications (ICoDSA), pp. 1–5, 2020.
12. A. Lin and W. Sun, *Multi-leader comprehensive learning particle swarm optimization with adaptive mutation for economic load dispatch problems*, Energies, vol. 12, no. 1, 116, 2019.
13. U. Mumtahina, S. Alahakoon, and P. Wolfs, *Hyperparameter tuning of load-forecasting models using metaheuristic optimization algorithms—a systematic review*, Mathematics, vol. 12, no. 21, 3353, 2024.
14. E. Cuevas, A. Trujillo, M. A. Navarro, and P. Diaz, *Comparison of recent metaheuristic algorithms for shape detection in images*, International Journal of Computational Intelligence Systems, vol. 13, no. 1, pp. 1059–1071, 2020.
15. J. Nayak, K. Vakula, D. Paidi, and B. Naik, *Spider monkey optimisation: state of the art and advances*, International Journal of Swarm Intelligence, vol. 4, no. 2, pp. 175–198, 2019.
16. D. Kumari, A. Sinha, S. Dutta, and P. Pranav, *Optimizing neural networks using spider monkey optimization algorithm for intrusion detection system*, Scientific Reports, vol. 14, no. 1, 17196, 2024.
17. D. J. Jose, C. H. Sulochana, and I. J. Mol, *Sustainable water quality prediction using adaptive spider monkey optimization with Bi-LSTM*, Earth Science Informatics, vol. 18, no. 3, 500, 2025.
18. A. Khan, A. Ali, J. Khan, F. Ullah, and M. Faheem, *Using Permutation-Based Feature Importance for Improved Machine Learning Model Performance at Reduced Costs*, IEEE Access, vol. 13, pp. 36421–36435, 2025, doi: 10.1109/ACCESS.2025.3544625.
19. H. Mandler and B. Weigand, *A review and benchmark of feature importance methods for neural networks*, ACM Computing Surveys, vol. 56, no. 12, pp. 1–30, 2024.
20. D. Lusiyanti, S. Anam, W. F. Mahmudy, and U. Sa'adah, *A Novel Hyperparameter Tuning Strategy Using Spider Monkey Optimization for Deep Neural Networks in Water Quality Prediction*, in 2025 International Electronics Symposium (IES), 2025, pp. 619–626.
21. M. Nachaoui, S. Lyaqini, and M. Chaouch, *Indicating if Water is Safe for Human Consumption Using an Enhanced Machine Learning Approach*, Statistics, Optimization & Information Computing, vol. 11, no. 1, pp. 70–81, 2023.
22. N. Khare, P. Devan, C. L. Chowdhary, S. Bhattacharya, G. Singh, S. Singh, and B. Yoon, *SMO-DNN: Spider Monkey Optimization and Deep Neural Network hybrid classifier model for intrusion detection*, Electronics, vol. 9, no. 4, 692, 2020.
23. C. Aarthi, V. J. Ramya, P. Falkowski-Gilski, and P. B. Divakarachari, *Balanced spider monkey optimization with Bi-LSTM for sustainable air quality prediction*, Sustainability, vol. 15, no. 2, 1637, 2023.
24. F. Eddaoudi, H. Lakhbab, and M. Naimi, *Benchmarking Metaheuristics for Neural Network Optimization: A Comprehensive Study*, Statistics, Optimization & Information Computing, vol. 15, no. 6, pp. 4922–4943, 2026.
25. B. Sahu, A. Panigrahi, B. Dash, P. K. Sharma, and A. Pati, *A hybrid wrapper spider monkey optimization-simulated annealing model for optimal feature selection*, International Journal of Reconfigurable and Embedded Systems, vol. 12, no. 3, p. 360, 2023.
26. S. H. A. Al-taee, B. H. Al-miyahi, M. J. Hamoud, W. A. M. Al-Jawher, H. Najm, and M. S. Mahdi, *A Hybrid COA-CNN Framework for Credit Card Fraud Detection Using Hyperparameter Optimization and Metaheuristic Feature Selection*, Statistics, Optimization & Information Computing, vol. 16, no. 1, pp. 33–48, 2026.
27. T. J. Mugejo and W. Govere, *Integrated ECOD-KNN Algorithm for Missing Values Imputation in Datasets: Outlier Removal*, in *International Journal of Innovative Science and Research Technology*, vol. 2024, 2024.
28. M. Shantal, Z. Othman, and A. A. Bakar, *A Novel Approach for Data Feature Weighting Using Correlation Coefficients and Min–Max Normalization*, in *Symmetry*, vol. 2023, 2023.
29. C. Sruneethi and B. Rajesh, *Intelligent Water Quality Assessment: Predictive Modeling for Potable Water Using Advanced Machine Learning Techniques*, in *International Journal of Scientific Research in Science and Technology*, vol. 2025, 2025.
30. G. Özsezer and G. Mermer, *Prediction of Drinking Water Quality With Machine Learning Models: A Public Health Nursing Approach*, in *Public Health Nursing*, vol. 2023, 2023.
31. J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, *Spider Monkey Optimization Algorithm for Numerical Optimization*, Memetic Comput., vol. 6, no. 1, pp. 31–47, 2014.

32. B. H. Pansambal and A. B. Nandgaokar, *Integrating Dropout Regularization Technique at Different Layers to Improve the Performance of Neural Networks*, in *International Journal of Advanced Computer Science and Applications*, vol. 2023, 2023.
33. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
34. J. Kennedy and R. Eberhart, *Particle Swarm Optimization*, in Proc. ICNN'95 - Int. Conf. Neural Networks, 1995, pp. 1942–1948, vol. 4.
35. S. Mirjalili, S. M. Mirjalili, and A. Lewis, *Grey Wolf Optimizer*, Adv. Eng. Softw., vol. 69, pp. 46–61, 2014.
36. X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2010.
37. Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*, in *Neural Networks: Tricks of the Trade*, Springer, Berlin, 2012, pp. 437–478.
38. D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2017.
39. L. Breiman, *Random Forests*, Machine Learning, vol. 45, pp. 5–32, 2001.
40. T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
41. World Health Organization, *Guidelines for Drinking-water Quality: Fourth Edition Incorporating the First and Second Addenda*, World Health Organization, Geneva, 2022.