Asian Journal of Water, Environment and Pollution. Vol. 22, No. 5 (2025), pp. 65-79.
doi: 10.36922/AJWEP025210165

ORIGINAL RESEARCH ARTICLE

Hybrid Nesterov-accelerated adaptive moment estimation–differential evolution optimization for long short-term memory-based dissolved oxygen prediction in water quality assessment

Tu Jun*, Azman Yasin, and Nur Suhaili Mansor
School of Computing, College of Arts and Sciences, University Utara Malaysia, Sintok, Kedah, Malaysia
***Corresponding author:** Tu Jun (tujun792324486@gmail.com)

Received: May 21, 2025; 1st revised: May 30, 2025; 2nd revised: June 13, 2025; Accepted: June 19, 2025; Published online: July 10, 2025

Abstract: Accurate and dynamic prediction of water quality indicators is increasingly critical due to rising pollution and water resource insecurity, particularly when dealing with high-dimensional, nonlinear time series data. Dissolved oxygen (DO), a key indicator of aquatic ecosystem health and pollution, requires high prediction accuracy for effective environmental management. This study aims to enhance the accuracy and adaptability of DO prediction by addressing the limitations of traditional deep learning methods, such as slow convergence and local optima. We propose a novel hybrid optimization framework that combines Nesterov-accelerated Adaptive Moment Estimation (Nadam) with the differential evolution algorithm. A dual-population cooperation strategy and an information exchange mechanism were incorporated during the training of a long short-term memory (LSTM) network to achieve a dynamic balance between global exploration and local exploitation. This improves the model's optimization efficiency and generalization. The research utilized a multivariate water quality time series dataset from Kaggle based on official data monitoring. Correlation analysis was conducted to ensure the scientific validity and effectiveness of the selected input variables. Experimental results demonstrated that the proposed method significantly outperforms traditional optimization strategies for DO prediction. Compared to the original Nadam optimizer, it reduced the mean squared prediction error by 47.8%, exhibiting enhanced adaptability and robustness in complex pollution scenarios. This study presents an effective optimization strategy to improve LSTM performance in water quality forecasting, along with a scalable and interpretable intelligent analysis framework. It provides both theoretical and practical support for water quality forecasting, early warning systems, and intelligent environmental monitoring.

Keywords: Water quality management; Dissolved oxygen prediction; Hybrid optimization; Nadam; Differential evolution

1. Introduction

The accelerating pace of global industrialization and urbanization has led to increasingly severe water quality deterioration, posing significant threats to aquatic ecosystems, biodiversity, and public health.[1,2] Among various water quality indicators, dissolved oxygen (DO) is widely recognized as a core variable for assessing the health of water bodies,[3] due to its critical role in regulating processes such as ecosystem respiration, organic matter

decomposition, and nutrient cycling. Hypoxic or anoxic conditions can cause aquatic organism mortality and even ecosystem collapse. Consequently, developing a high-precision, real-time DO prediction method is of paramount importance for pollution monitoring, water resource management, and early warning systems.[4]

In recent years, deep learning methods have gained significant attention in environmental modeling, demonstrating exceptional capabilities in capturing nonlinear relationships, particularly within hydrological time series analysis.[5,6] Among these, long short-term memory (LSTM) networks excel at capturing temporal dependencies and have been successfully applied in various environmental forecasting scenarios.[7] However, their training process is highly sensitive to optimization algorithms. Traditional gradient-based optimizers (e.g., stochastic gradient descent, Adam) often converge prematurely or become trapped in local optima when applied to high-dimensional, noisy environmental datasets, thereby limiting model generalization and prediction accuracy.[8]

To address these limitations, this study proposes a novel hybrid optimization strategy combining Nesterov-accelerated Adaptive Moment Estimation (Nadam) and differential evolution (DE) to enhance the training stability and global search capability of LSTM models for DO prediction tasks. Nadam enhances local convergence by incorporating a momentum-based mechanism,[9] while DE, a population-based stochastic algorithm, enhances global exploration through evolutionary and mutation mechanisms.[10] A key innovation of this study is the introduction of an information exchange mechanism (IEM), which enables the Nadam and DE sub-populations to share elite solutions during training, thereby achieving a dynamic balance between local exploitation and global exploration.

Beyond improving DO prediction accuracy, the proposed method offers broad applicability for environmental modeling. By providing early signals of pollution changes and ecological stress, accurate DO trend modeling can inform the optimization of water treatment strategies, particularly in identifying pathways influenced by metal, inorganic, and organic pollutants.[11,12] Therefore, this study not only extends the application boundaries of deep learning in complex time series but also provides a novel decision-support tool for intelligent pollution control and sustainable water resource management.

In summary, the main contributions of this study are: (i) proposing a hybrid Nadam –DE optimization framework for training LSTM models; (ii) designing and introducing an IEM to enhance optimization stability and prediction robustness; and (iii) validating the method's superior performance on real-world DO prediction tasks. The structure of the article is as follows: Section 1 presents the research background and motivation (Introduction); Section 2 reviews relevant literature and the current state of research (Literature Review); Section 3 describes the materials and methods employed in the study (Materials and Methods); Section 4 details the experimental design and results (Results); Section 5 discusses the findings (Discussion); Section 6 explores practical application prospects of this research (Practical Applications); and Section 7 summarizes the paper and highlights key contributions and future research directions (Conclusions).

2. Literature review

In recent years, DO prediction has become a vital component of water quality management and ecosystem health monitoring. Researchers have extensively explored data-driven approaches, including support vector machines (SVMs), artificial neural networks, and fuzzy linear regression, to address the nonlinearity and data scarcity inherent in aquatic systems. For example, Ji *et al.*[13] accurately predicted DO concentration in a low-oxygen river system using an SVM model in the Wenruitang River in China. Khan and Valeo[14] developed a fuzzy regression framework to handle environmental uncertainties, outperforming conventional regression techniques by robustly managing sparse datasets and systemic complexity. Wu *et al.*[15] integrated sliding window techniques with particle swarm optimization and backpropagation neural networks to enhance DO prediction accuracy. Malek *et al*.[16] applied an SVM model to analyze data from two large lakes in Malaysia, using the forward selection method to optimize input parameters and achieving a prediction accuracy of 74%.

Although traditional forecasting methods, such as SVM, show high accuracy with nonlinear problems, they have limitations when handling long-term and complex time series data. Specifically, they require a large number of feature selections and cannot adequately capture dynamic changes in time series. To overcome these challenges, LSTM has been introduced for DO prediction in recent years. Its unique memory unit structure effectively captures long-term dependencies in DO time series data, improving prediction accuracy. For example, Tan *et al*.[17] combined principal component analysis with LSTM, achieving a 2.71% reduction in mean squared error (MSE) and a 9.03% improvement in

mean absolute percentage error compared to standalone LSTM models. Hao[18] developed a DO prediction method combining the maximum information coefficient with LSTM neural networks to identify the most influential environmental variables. This model outperformed traditional statistical methods and other machine learning techniques in predictive accuracy. However, LSTM models often suffer from slow convergence during training. To accelerate this process, many researchers employ the Nadam algorithm. For instance, Gandh *et al.*[19] designed a hybrid LSTM–gated recurrent unit architecture for intelligent aquaculture water quality forecasting. Comparative evaluations of optimizers (e.g., Adaptive Moment Estimation [Adam], Nadam) revealed enhanced prediction accuracy and computational efficiency across two distinct water quality datasets. Lee *et al*.[20] utilized Adam-optimized LSTM to impute missing DO values in land-based aquaculture systems, achieving an error margin of ~3.25%. Nevertheless, gradient-based methods like stochastic gradient descent may converge to suboptimal local minima in the non-convex optimization landscape of LSTM training. To address this, Shuai and Tian[21] applied DE to refine LSTM for high-speed rail passenger flow prediction, reducing error rates by 2.3988% relative to baseline LSTM. Similarly, Peng *et al.*[22] deployed DE-optimized LSTM for electricity price forecasting, outperforming competing models in multi-task scenarios. Although DE excels in global search and local optimization, it is less effective than gradient optimization in local development.

Compared to the Adam optimizer, Nadam incorporates Nesterov momentum, enabling anticipation of the gradient direction during parameter updates to accelerate convergence and reduce oscillation. In addition, Nadam is more stable during training, especially when dealing with sparse gradients and dynamic data. However, no prior work has combined Nadam with DE to synergize global exploration and local exploitation, thereby enhancing model optimization. To bridge this gap, this study proposes Nadam–DE, a hybrid algorithm leveraging Nadam's rapid local convergence and DE's global search capabilities for aquatic DO prediction. This framework strategically balances exploration and exploitation, improving both prediction accuracy and training efficiency.

3. Materials and methods

3.1. Basic principles of Nadam and DE algorithm

3.1.1. Nadam

In the field of optimization algorithms, Nadam and DE[23] represent typical paradigms of gradient-based and population-based optimization strategies. Nadam is an improved method that introduces the Nesterov momentum mechanism into the Adam optimizer, aiming to enhance the predictive ability and convergence efficiency of parameter updates. The core idea is to simultaneously utilize first-order and second-order moment estimations to smooth the gradient information and predict the future gradient direction through Nesterov momentum, resulting in better convergence trends and robustness during parameter updates.[9]

In its parameter update mechanism, the gradient is first subjected to first-order moment estimation (i.e., momentum term) and second-order moment estimation (i.e., variance term), calculated as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t \quad \text{(I)}$$

Where g_t represents the current gradient and β_1 is the momentum decay factor typically set to 0.9. The second moment is calculated as:

$$v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^2 \quad \text{(II)}$$

Where β_2 is the attenuation factor for the second moment, usually set to 0.999, and is used to estimate the squared mean of the gradient. To avoid deviation, Nadam corrects these estimates and introduces a modified form of Nesterov momentum:

$$\hat{m}_t = \beta_1 m_t + \left(1-\beta_1\right) g_t \quad \text{(III)}$$

The final parameter update formula is:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\beta_1 \hat{m}_t + \left(1-\beta_1\right)\frac{g_t}{1-\beta_1^t}}{\sqrt{\hat{v}_t} + \varepsilon} \quad \text{(IV)}$$

Where η denotes the learning rate, ϵ represents a numerical stabilizer to avoid zero-division errors (e.g., 1×10^{-8}), and $\hat{v}_t = v_t / \left(1-\beta_2^t\right)$ is the corrected second moment estimate. Nadam offers advantages such as fast convergence and strong adaptability, making it especially suitable for gradient dilute sulfur or high-dimensional continuous space problems. However, it relies on gradient information and is susceptible to becoming trapped in local optima when faced with complex loss functions.

3.1.2. DE

In contrast, the DE algorithm is a population-based, heuristic global optimization method that evolves better solutions through the simulation of mutation,

crossover, and selection operations inspired by natural selection. In the mutation stage, DE perturbs the current individual using difference vectors randomly selected from the population to form an exploratory mutation vector. During the crossover stage, the mutation vector is combined with the current individual according to a specified probability, ensuring diversity in the search process. In the selection stage, the algorithm retains individuals with better fitness values to proceed to the next generation. This strategy enables DE to exhibit strong global search capabilities without relying on gradient information, making it particularly suitable for complex, non-convex, or non-differentiable problems.

However, the local development ability of DE is relatively weak.[24] Once the solution space begins to converge, it often lacks a fine search mechanism. Moreover, the computational cost associated with population-based evolution can be relatively high. Specifically, in the mutation stage, three distinct individuals, x_{r1}, x_{r2}, x_{r3} are selected from the population to construct a difference vector. The mutation vector is generated as follows:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{V}$$

Where F∈ [0,2] is the scaling factor that controls the amplitude of the differential disturbance. In the crossover operation, the mutation vector is mixed with the current individual x_i using a crossover probability CR∈[0,1] to produce the test vector u_i:

$$u_{i,j} = \begin{cases} v_{i,j}, & if\ rand \le CR\ or\ j = j_{\text{rand}} \\ x_{i,j}, & otherwise \end{cases} \tag{VI}$$

Here, j_{rand} is a randomly selected index that ensures that at least one dimension is inherited from the mutation vector. Finally, u_i and x_i are compared using the fitness function. The individuals with better performance are retained for the next generation:

$$x_i^{t+1} = \begin{cases} u_i, & if\ f(u_i) \le f(x_i) \\ x_i, & otherwise \end{cases} \tag{VII}$$

The DE algorithm is widely used in various complex optimization problems due to its simple structure and flexible implementation. It performs particularly well in scenarios where the search space is complex and gradient information is difficult to obtain. However, the pure DE method has limited local search ability, and its computational cost can be relatively high due to the population evolution process, especially in high-dimensional spaces.

In summary, Nadam and DE exhibit complementary optimization characteristics. Nadam is suitable for continuous optimization problems with known gradients and offers excellent local convergence speed, while DE is capable of conducting large-scale global searches without requiring gradient data. Combining the two allows for the generation of high-quality global initial solutions via DE, followed by accelerated local convergence in the solution space through Nadam. This hybrid strategy improves both the accuracy and efficiency of the overall optimization process. It enhances robustness, reduces the risk of becoming trapped in local optima, and is of great significance for solving complex, high-dimensional optimization problems.

3.2. Design of hybrid algorithms and IEM

To fully leverage the strengths of Nadam and DE, this study proposes a hybrid optimization algorithm that integrates the fast local convergence capability of Nadam with the strong global search ability of DE. In this framework, the overall population was divided into two subpopulations: one employed the Nadam algorithm for gradient-driven local development, while the other applied the DE algorithm to perform global exploration. To facilitate dynamic collaboration between the two, an IEM was introduced to regularly exchange optimal solutions between subpopulations during the algorithm operation process, promoting effective integration of global and local search strategies.

The core idea of this IEM lies in a strategy termed "exchanging the superior for the inferior." Specifically, at regular intervals (for example, every five generations), the current optimal individual is exchanged between the two subpopulations. This promotes information flow across the solution space, prevents the algorithm from converging to local optima, and simultaneously enhances both search diversity and accuracy. The basic workflow of the IEM is shown in Figure 1.

Within each exchange cycle, the individual with the current best performance (i.e., the solution with the lowest objective function value) was first selected from the Nadam subpopulation. This elite individual replaced the poorest-performing individual in the DE subpopulation. This step ensures that the DE subpopulation can inherit the local region optimization achievements obtained by Nadam, enabling it to continue global search and mining in this region.

Conversely, the best-performing individual from the DE subpopulation was selected to replace the worst-performing individual in the Nadam subpopulation.

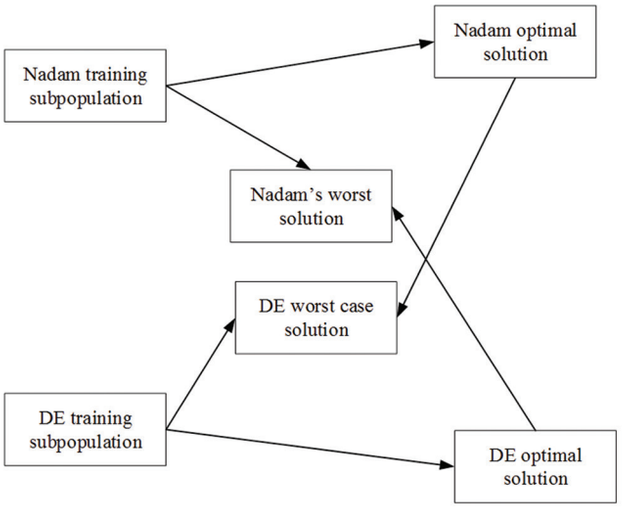

Figure 1. Information exchange mechanism
Abbreviations: DE: Differential evolution; Nadam: Nesterov-accelerated adaptive moment estimation.

This enables the global search results of DE to guide Nadam's local optimization path. By utilizing the globally optimal solution identified by DE as a new starting point, Nadam can perform high-precision, gradient-based fine-tuning. This design further reduces the error in the solution and improves the overall convergence quality and optimization efficiency.

The entire information exchange process not only maintains the diversity of the algorithm but also mitigates the risk of subpopulations becoming trapped in their respective local optima. Through this two-way information flow and collaborative optimization strategy, the hybrid algorithm achieved an effective balance between exploration and development. This approach aligns with the optimization principle of "collaborative parallelism of local and global search." Parameters of this mechanism (such as the exchange cycle frequency and replacement ratio) can be flexibly adjusted according to the complexity of specific problems to accommodate different optimization requirements.

Through the design of the IEM, Nadam and DE no longer operate independently in parallel but form a collaborative system with a coupled feedback mechanism. Nadam efficiently converges in the local space using gradient information, while DE continuously explores the broader solution space and provides high-quality search directions. By exchanging the information of optimal individuals, the two algorithms complement each other's strengths and compensate for their weaknesses. This hybrid mechanism helps overcome the performance bottlenecks typically faced by a single optimization algorithm. Experimental results also confirm that this approach achieves faster convergence speed and higher solution quality compared across multiple benchmark function tests, demonstrating its practical effectiveness for solving complex, high-dimensional optimization problems.

3.3. Process and characteristics of the hybrid algorithm

In high-dimensional complex optimization problems, a single optimization strategy often presents a trade-off between search ability and convergence efficiency. Local gradient optimization methods (such as Nadam) offer rapid convergence but are prone to becoming trapped in local optima. In contrast, swarm intelligence algorithms (such as DE) possess good global search capabilities but often suffer from reduced optimization accuracy due to insufficient local development capabilities. To address this contradiction, this study proposes a hybrid optimization algorithm based on the coevolution of twin populations. By effectively integrating Nadam and DE, an optimization framework was constructed that combines global exploration with local development.

The core of this algorithm lies in dividing the population into two functional sub-modules: the local population, optimized using Nadam, and the global population, optimized using DE. A periodic IEM enabled dynamic coupling between the two search strategies, enabling the optimization process to maintain both global search capabilities and local convergence accuracy. The optimization process of the hybrid algorithm can be divided into five stages: initialization, local development, global search, information exchange, and termination determination. The specific steps are detailed in the following subsections.

3.3.1. Population initialization and structure setting

At the beginning of the algorithm, two heterogeneous subpopulations were initialized in the solution space. Let the total population size be N, divided into the local subpopulation P_L and the global subpopulation P_G, with the number of individuals being N_L and N_G, satisfying:

$$N = N_L + N_G,\ P_L = \left\{x_i^L\right\}_{i=1}^{N_L},\ P_G = \left\{x_j^G\right\}_{j=1}^{N_G} \quad \text{(VIII)}$$

Here, each individual $x \in R^d$ represents a candidate solution with dimension d. During the initialization stage, the initial positions of all individuals were randomly generated through uniform distribution within the given search boundary:

$$x_i^L(0), x_j^G(0) \sim \mathcal{U}\left(x_{\min}, x_{\max}\right) \quad \text{(IX)}$$

For individuals in the local subpopulation, it was also necessary to initialize the first and second moment estimates of their gradients:

$$m_0^{(i)} = 0,\ v_0^{(i)} = 0 \quad \text{(X)}$$

This setting provides the initial conditions for the subsequent parameter updates of the Nadam optimizer.

3.3.2. Local optimization stage

The evolution of the local subpopulation P_L mainly relies on the gradient-driven mechanism of the Nadam optimizer. For each generation *t*, the current gradient was calculated as $g_t^{(i)} = \nabla f\left(x_i^L(t)\right)$, followed by multi-order estimation and parameter updates based on this gradient:

(a) First-order moment estimation (momentum):

$$m_t^{(i)} = \beta_1 m_{t-1}^{(i)} + \left(1-\beta_1\right) g_t^{(i)} \quad \text{(XI)}$$

(b) Second-order moment estimation (variance):

$$v_t^{(i)} = \beta_2 v_{t-1}^{(i)} + \left(1-\beta_2\right)\left(g_t^{(i)}\right)^2 \quad \text{(XII)}$$

(c) First-order moment Nesterov correction:

$$\hat{m}_t^{(i)} = \beta_1 m_t^{(i)} + \left(1-\beta_1\right) g_t^{(i)} \quad \text{(XIII)}$$

(d) Second-order moment deviation correction:

$$\hat{v}_t^{(i)} = \frac{v_t^{(i)}}{1-\beta_2^t} \quad \text{(XIV)}$$

(e) Parameter update formula:

$$x_i^L(t+1) = x_i^L(t) - \eta \cdot \frac{\hat{m}_t^{(i)}}{\sqrt{\hat{v}_t^{(i)}} + \varepsilon} \quad \text{(XV)}$$

Here, $\beta 1$ and $\beta 2$ control the decay rates of momentum and variance, η is the learning rate, and ϵ is a small constant added to avoid a division by zero.

Through the above-mentioned gradient dominance mechanism, the local subpopulations can rapidly converge toward the target region in the search space. This approach is especially suitable for problems where the shape of the function is known or where gradient information is available.

3.3.3. Global search stage

The global subpopulation P_G adopts the DE algorithm for iteration to ensure the wide coverage of the search space. For each individual x_j^G, its evolutionary process includes the following three steps:

(a) Differential variation

Randomly select three different individuals, $x_{r1}, x_{r2}, x_{r3} \in P_G \setminus \left\{x_j^G\right\}$, and construct the mutation vector:

$$v_j = x_{r1} + F\cdot(x_{r2}\text{-}x_{r3}) \quad \text{(XVI)}$$

Where $F \in [0,2]$ is the scaling factor.

(b) Cross-operation

Construct the test vector using the cross probability $\mathrm{CR} \in [0,1]$:

$$u_{j,k} = \begin{cases} v_{j,k}, & if\ rand\ (0,1) \le CR\ or\ k = k_{\text{rand}} \\ x_{j,k}^G, & otherwise \end{cases} \quad \text{(XVII)}$$

Here, k_{rand} ensures that at least one dimension is taken from the mutation vector.

(c) Selection operation

Compare u_j and x_j^G based on the fitness function $\mathrm{f}(\cdot)$:

$$x_j^G(t+1) = \begin{cases} u_j, & if\ f\left(u_j\right) \le f\left(x_j^G\right) \\ x_j^G, & otherwise \end{cases} \quad \text{(XVIII)}$$

The evolutionary operation of DE does not rely on gradients and is suitable for optimization problems with complex search spaces and non-differentiable or discontinuous objective functions.

(d) IEM

To achieve synergy between the two optimization strategies, this study designed an IEM based on the principle that "the superior replaces the inferior." This mechanism was executed once every T_{ex} proxy. The specific process is as follows:

Select the best-performing individual from the Nadam subpopulation P_L:

$$x_L^* = \arg\min_{x \in P_L} f(x) \quad \text{(XIX)}$$

Select the worst-performing individual from the DE subpopulation P_G:

$$x_G^{worst} = \arg\max_{x \in P_G} f(x) \quad \text{(XX)}$$

Replace x_G^{worst} with x_L^* to make the DE subpopulation inherit the local optimal information.

Similarly, extract the current global optimal solution from P_G:

$$x_G^* = \arg\min_{x \in P_G} f(x) \quad \text{(XXI)}$$

Use it to replace the worst-performing individual x_L^{worst} in the Nadam subpopulation, injecting the global search results into the local search path.

This two-way interaction mechanism not only promotes the flow of high-quality information in the understanding space but also helps overcome the limitations of local optimality, thereby improving overall search efficiency and solution accuracy.

(e) Termination criteria and optimal solution output

The hybrid algorithm terminated when either of the following conditions was met: The maximum number of iterations T_{max} was reached, or the fitness value of the optimal individual in the entire population fell below the predefined threshold δ.

The final optimization result was recorded as:

$$x^* = \arg \min_{x \in P_L \cup P_G} f(x) \quad \text{(XXII)}$$

3.4. Application of the hybrid optimization algorithm in LSTM training

To effectively improve the generalization performance and convergence accuracy of deep learning models in complex water quality data prediction tasks, this study introduced the previously constructed Nadam–DE hybrid optimization algorithm into the training process of the LSTM network. Unlike the traditional training mechanism based on back propagation and a single gradient optimizer, the hybrid optimization strategy emphasizes a wide-area search in the global parameter space during the early stage of training and focuses on gradient-driven fine development in the middle and later stages. This approach achieved the optimization goal of dynamic collaboration between "exploration" and "development" in the LSTM parameter learning process.

3.4.1. Parameter encoding and search space definition

Under this hybrid optimization framework, all trainable parameters of the LSTM model were treated as variables to be optimized. These parameters were vectorized and encoded as continuous, real-valued individuals. Assuming the model structure is fixed, the parameter set can be expressed as:

$$\theta = \{W_{xi}, W_{hi}, b_i, W_{xf}, W_{hf}, b_f, W_{xc}, W_{hc}, b_c, W_{xo}, W_{ho}, b_o\}$$

Here, W represents the weight matrices corresponding to each gating mechanism (input gate, forget gate, candidate memory unit, and output gate), and b denotes the bias term. After flattening the weights of ownership and the bias term into vectors, they collectively form a single individual $\theta_i \in R^d$ in the hybrid algorithm, where d is the total dimensionality of the parameters.

3.4.2. Construction of the loss function and fitness evaluation

The hybrid algorithm drove the evolution process by evaluating the predictive ability of each individual under the current LSTM model. Specifically, the individual parameter vector θ_i was loaded into the network, the model performed forward propagation on the training data, calculated the prediction result $\hat{y}_t$, and used the MSE as the fitness function:

$$\mathcal{L}(\theta_i) = \frac{1}{N} \sum_{t=1}^{N} \left(y_t - \hat{y}_t \right)^2 \quad \text{(XXIII)}$$

Here, y_t is the true label, $\hat{y}_t$ is the prediction result of the model under the parameter θ_i, and N is the sample size.

3.4.3. Embedding process of the hybrid optimization algorithm

Embedding the hybrid optimization algorithm into the parameter training of the LSTM model, the training process can be regarded as a complex black-box function optimization problem. The specific process is as follows:

(a) Initialization stage: Generate the initial population $\{\theta_1, \theta_2, ..., \theta_P\}$, which is divided into the Nadam subpopulation and the DE subpopulation. Local gradient updates and global mutation searches are then performed.

(b) Parallel optimization stage: In each generation, the Nadam subpopulation uses gradient information to perform first-order and second-order moment estimates for individuals and updates the parameters. Meanwhile, the DE subpopulation completes global exploration through differential variation and crossover mechanisms.

(c) Fitness assessment: For each individual, perform forward propagation in the LSTM model, calculate the loss $L(\theta_i)$, and adjust the ranking of individuals based on their fitness.

(d) IEM: At intervals of T_{swap} generations, individual replacements based on the principle of "superior replacing inferior" are carried out between subpopulations. The Nadam subpopulation injects its current optimal individual into the DE subpopulation, and vice versa, promoting information sharing and collaborative search.

(e) Convergence determination: When the value of the optimal loss function tends to stabilize or reaches the maximum iterative algebra G_{max}, the optimization

terminates, and the global optimal parameter θ^* is output for model deployment.

4. Experiment and result analysis

4.1. Dataset and experimental settings

The water quality dataset used in this experiment was obtained from publicly available secondary data on Kaggle. The water quality information in the dataset was regularly collected by volunteers under the supervision of the management department of the protected area. Sampling was carried out biweekly at designated water body locations within the protected area to ensure coverage across the spatial range of different water bodies. This dataset has been meticulously maintained over an extended period, ensuring the accuracy and reliability of the monitoring results. It serves as a high-quality source of water quality monitoring data.

The dataset contains multiple water quality parameters, including DO, water temperature, pH, turbidity, electrical conductivity, and others. The DO concentration was the target variable for prediction, and other parameters were used as input features to train the LSTM model. Sample entries are shown in Table 1.

4.1.1. Data normalization and correlation analysis

In this study, Min-Max normalization was used to standardize the dataset, eliminating differences across eigenvalue dimensions and ensuring the stability of model training. The normalization formula is as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{(XXIV)}$$

The data samples after normalization are presented in Table 2.

In this study, to investigate the influence of different water quality parameters on DO, we employed scatter plot analysis to examine the relationships between pH, temperature, turbidity, conductivity, and DO. Scatter plots serve as an intuitive visualization tool, enabling the preliminary identification of linear or nonlinear associations between variables.[25] This analysis provides valuable insights for subsequent complex modeling efforts, particularly in determining which features may exhibit linear or nonlinear dependencies with DO. The results of the scatter plot analysis are presented in Figure 2.

From the scatter plot results, pH and DO demonstrate a somewhat positive correlation, though the relationship is not strictly linear, suggesting the presence of potential nonlinear patterns. Temperature and DO show no clear linear or nonlinear trend, with scattered data points indicating a complex and ambiguous influence of temperature on DO. The relationship between turbidity and DO appears weak, with data points exhibiting a near-random distribution, implying that turbidity has no significant impact on DO. In contrast, conductivity and DO exhibit a strong positive correlation with a relatively linear relationship, confirming conductivity as a key predictor for DO.

These preliminary findings offer valuable guidance for our subsequent modeling work. Integrating these

Table 1. Dataset sample

Sample ID	pH	Temperature (°C)	Turbidity (NTU)	Dissolved oxygen (mg/L)	Conductivity (S/cm)
1	7.25	23.1	4.5	7.8	342
2	7.03	21.5	3.9	8.3	356
3	7.38	22.9	3.2	9.5	327
4	7.45	20.7	3.8	8.1	352
5	7.19	21.2	4.2	8.8	350

Table 2. Standardized data sample

Sample ID	pH	Temperature (°C)	Turbidity (NTU)	Dissolved oxygen (mg/L)	Conductivity (S/cm)
1	0.646154	0.848485	0.7	0.461538	0.481481
2	0.307692	0.363636	0.4	0.589744	0.740741
3	0.846154	0.787879	0.05	0.897436	0.203704
4	0.953846	0.121212	0.35	0.538462	0.666667
5	0.553846	0.272727	0.55	0.717949	0.62963

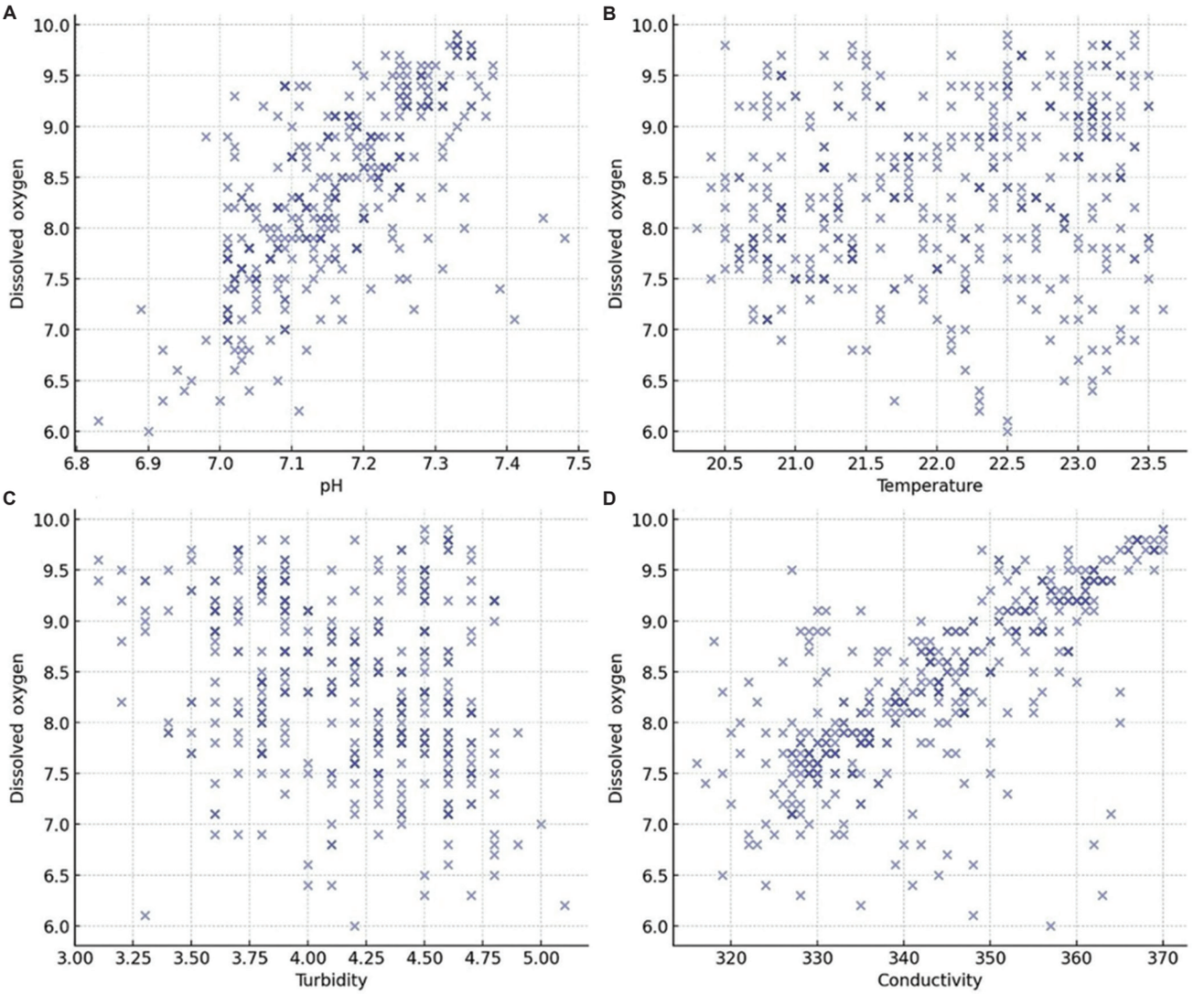

Figure 2. Scatter plots of water quality parameters versus dissolved oxygen. (A) pH versus dissolved oxygen; (B) Temperature versus dissolved oxygen; (C) Turbidity versus dissolved oxygen; (D) Conductivity versus dissolved oxygen.

features into a multi-feature predictive model allows for the exploration of both individual effects and their potential. Advanced machine learning models–such as Random Forest, XGBoost, and LSTM–are particularly well-suited for this task, as they can automatically capture complex linear and nonlinear dependencies among features.[26] Therefore, by leveraging multi-feature integration and the learning capabilities of these algorithms, we anticipate improved accuracy in forecasting DO levels.

4.1.2. Characteristic importance analysis

To gain a deeper understanding of the contribution of various water quality indicators in predicting DO, this study developed a prediction model based on an LSTM network and employed the permutation feature importance method to evaluate the importance of input features.[27] This method involves independently shuffling the time series of each feature and observing the change in model performance (measured by MSE), thereby assessing the actual impact and contribution of each feature to the prediction outcomes within the specific model.

During the evaluation, the baseline MSE of the model on the original test set was 0.027023. Subsequently, five repeated random shuffling experiments were conducted for each feature. The average increase in MSE and its standard deviation were calculated as indicators of feature importance. The results are shown in Figure 3.

The "MSE increase" and "standard deviation (Std)" values for each parameter are presented in Table 3. "MSE increase" refers to the average increase in the model's prediction error, compared to the original data, after shuffling a specific feature. "Std" indicates the variability in the MSE increase across multiple shuffling repetitions. A smaller standard deviation suggests a more stable and reliable assessment of feature importance, while a larger standard deviation indicates greater uncertainty in the evaluation due to varying impacts across shuffles.

The results reveal that among the four input features, conductivity has the most significant impact on DO prediction, with an average MSE increase of 0.005783 after shuffling, substantially higher than that of the other features. pH ranks second, indicating its strong predictive contribution in reflecting the chemical properties of water. In contrast, turbidity and temperature show lower importance, exerting minimal influence on model performance.

During experimentation, the dataset was partitioned into training and testing subsets using an 80:20 ratio. Model training employed the proposed hybrid optimization strategy integrating Nadam and DE, and was benchmarked against standalone Nadam and DE

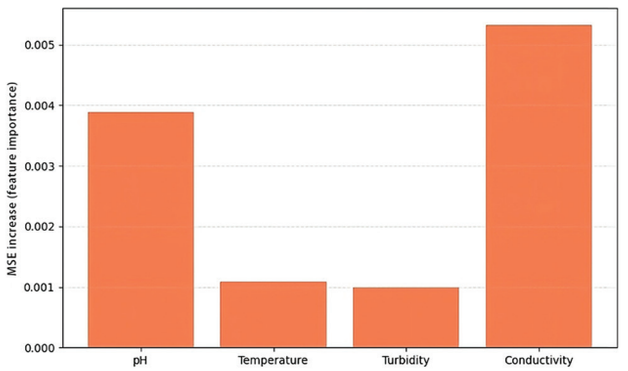

Figure 3. Feature importance (permutation method) for dissolved oxygen prediction

Table 3. Evaluation results of input feature importance

Feature	MSE increase	Standard deviation
pH	0.003135	0.000362
Temperature	0.001345	0.000523
Turbidity	0.000697	0.001291
Conductivity	0.005783	0.000875

Abbreviations: MSE: Mean squared error; Std: Standard deviation.

implementations. The computational framework utilized Python[28] and the Keras deep learning framework,[29] with GPU acceleration enabled for training.

4.2. Model settings

The neural network comprises an LSTM layer for temporal feature extraction and a dense layer for regression output.[30] Critical hyperparameters, including the number of LSTM units and the learning rate, were optimized using an evolutionary approach.[31] MSE[32] was employed as the loss metric to evaluate regression performance.

The DE algorithm optimized hyperparameters by simulating selection, mutation, and crossover processes inspired by natural evolution. As shown in Table 4, the search range for the learning rate was set to [$1e^{-4}$, $1e^{-2}$], while the number of LSTM units ranged from 10 to 100. The population size was set to 15, with a single iteration per generation. The entire optimization process spanned 20 generations of evolution, with the DE population being updated once per generation. Ultimately, in each generation, the DE algorithm selected the individual with the lowest verification loss as the global optimal solution of that generation.

In each generation, the Nadam optimization algorithm was used to train the model individuals generated by DE. The Nadam population size was set to 5, and each individual randomly generated a different learning rate and number of LSTM units. Nadam trained each individual and selected the best-performing one as the local optimal solution for that generation.

During the optimization process, every two generations, the optimal individuals from the DE and Nadam populations were exchanged. Specifically, the Nadam population received the globally optimal individual with the lowest verification loss from the DE population and replaced its worst-performing member. This ensured that the Nadam population could utilize

Table 4. Parameter settings

Parametric	Value
LSTM layer	10 – 100
Learning rate	$1e^{-4}$ – $1e^{-2}$
Generation	30
DE number of iterations	10
DE population size	15
Frequency of information exchange	Every 2 generations

Abbreviations: DE: Differential evolution; LSTM: Long short-term memory.

the global search results of DE. Conversely, the DE population received the local optimal solution from the Nadam population to enhance its performance in local search. This two-way IEM ensured that DE benefits from Nadam's local optimization results, while Nadam leverages DE's global search ability.

4.3. Result analysis

4.3.1. Training results for Nesterov-accelerated adaptive moment estimation–DE optimization algorithm

First, the LSTM model was trained using the Nadam–DE optimization algorithm. The optimized hyperparameter included a learning rate of $9.52e^{-3}$ and 79 LSTM units.

Based on these hyperparameters, the model was trained, achieving an MSE of 0.019278116524219513 after training. This result indicates that Nadam–DE can effectively improve the prediction performance of the model by globally searching for optimal hyperparameters. During training, changes in the loss function are shown in Figure 4, where both training and validation losses decrease steadily with the number of iterations and stabilize after several epochs. Nadam–DE showed good convergence in the optimization process.

4.3.2. Nadam training results

To further compare the effects, the model was also trained using only the Nadam optimizer, with the same learning rate ($9.52e^{-3}$) and number of LSTM units (79). Under these identical hyperparameter settings, the MSE achieved by Nadam after training was 0.036936987191438675, which is significantly higher than that obtained using Nadam–DE. This result suggests that although Nadam can accelerate local convergence, it is prone to becoming trapped in local optimization due to its lack of global optimization ability, ultimately affecting the final model accuracy.

During Nadam optimization, the training and verification loss changes are illustrated in Figure 5. Both losses decrease with the number of training epochs. However, compared to Nadam–DE, the verification loss converges more slowly and reaches a higher final value.

4.3.3. Algorithm comparison

To evaluate the effectiveness of the proposed Nadam–DE optimizer in time series forecasting tasks, we compared its performance with that of the standard Nadam optimizer using the MSE metric after training. Experimental results show that the post-training MSE for Nadam–DE is 0.0193, whereas for Nadam it is 0.0369. To quantify the magnitude of performance improvement, we calculated the percentage reduction in prediction error using the following formula:

$$\text{Error Reduction Rate} = \frac{\text{Original Error} - \text{Optimized Error}}{\text{Original Error}} \times 100\% \quad \text{(XXV)}$$

Substituting the actual values into the formula yields:

$$\frac{0.0369 - 0.0193}{0.0369} \times 100\% \approx 47.8\%$$

These results indicate that Nadam–DE reduces the prediction error by approximately 47.8% compared to Nadam. This significant performance improvement demonstrates that integrating the DE mechanism into the Nadam framework effectively enhances optimization efficiency and improves the model's generalization capability. As a result, the hybrid approach achieved

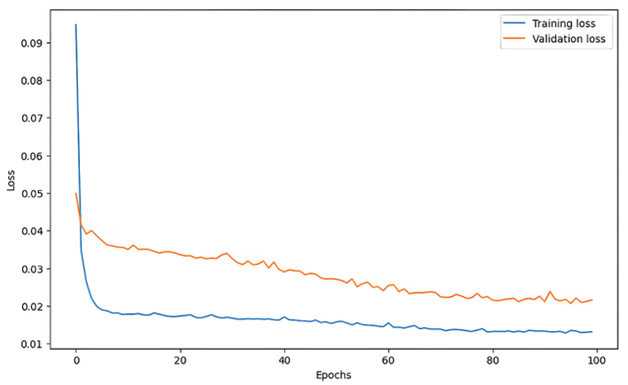

Figure 4. Loss function changes during training with Nesterov-accelerated adaptive moment estimation–differential evolution optimization algorithm

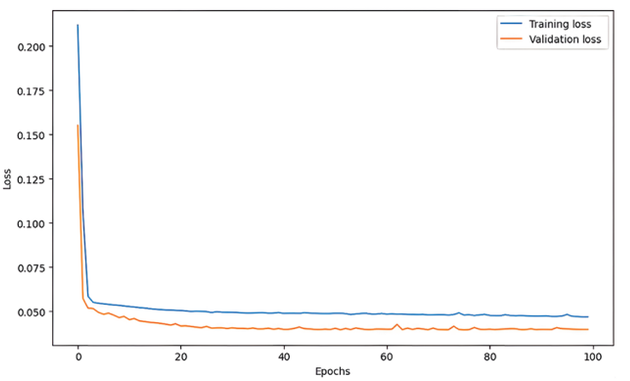

Figure 5. Loss function progression under Nesterov-accelerated adaptive moment estimation optimization

higher accuracy and greater training stability in complex, non-stationary time series forecasting tasks.

4.4. Model transferability verification

To further validate the applicability of the Nadam–DE algorithm, this study conducted experiments using an additional water quality dataset. This dataset was collected by Dr. J. Thad Scott in 2019 under the support of the Tarrant Regional Water District, with samples sourced from Eagle Mountain Lake. The data have been publicly released on the Kaggle platform. The loss curve during model training on this dataset is shown in Figure 6.

After model training and evaluation, the final MSE reached 0.015074709430336952, indicating a minimal error between the predicted and actual DO values, thereby demonstrating the model's accuracy. The model not only exhibits strong predictive capability on the original dataset but also maintains robust generalization performance and transferability under cross-source data conditions. This verification significantly enhances the practical viability of the model for diverse aquatic environments and provides a solid foundation for its deployment in real-world water quality monitoring and management tasks.

5. Discussion

To enhance the predictive ability of DO in the LSTM model, this study introduced a hybrid optimization algorithm, Nadam–DE, which combines the strengths of Nadam and DE. This method integrates the local optimization efficiency of Nadam with the global exploration capability of DE, augmented by an information-sharing mechanism to achieve balanced and dynamic optimization.

Experimental results demonstrated that, under identical hyperparameter configurations, the LSTM model optimized by Nadam–DE achieves an MSE of 0.0193. In comparison, using the Nadam optimizer alone yielded an MSE of 0.0369, representing an error reduction of approximately 47.8%. This highlights the significant advantages of the hybrid approach in terms of both convergence speed and prediction accuracy.

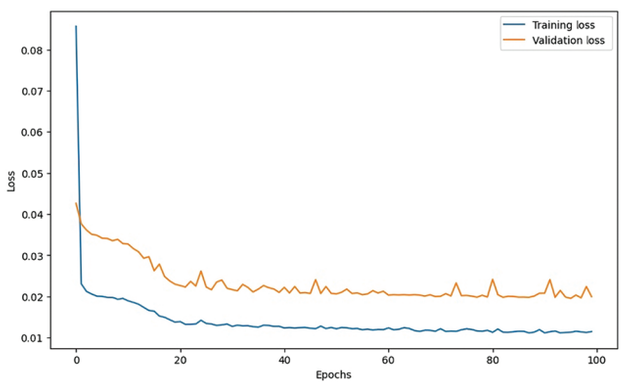

Figure 6. Loss function changes on cross-source dataset

However, an in-depth analysis of the convergence process reveals the presence of oscillations during the local iteration phase, suggesting potential instability in the information exchange frequency or strategy scheduling. Furthermore, although Nadam–DE has shown strong performance in retrospective prediction, its efficacy requires further validation in scenarios involving extended prediction lead times, extreme pollution events, and cross-watershed generalization capability.

Overall, the outstanding performance of this method in water quality prediction tasks underscores its significant practical application value. As a critical environmental parameter for aquatic ecosystems, DO is directly related to a water body's self-purification capacity and ecological balance. Prediction models optimized using Nadam–DE enable high-precision, dynamic monitoring and forecasting of DO concentrations. They also provide a scientific basis for decision-making in pollution early warning, water body remediation, and resource allocation. This is of great importance for improving the efficiency of water resource utilization, safeguarding ecological security, and advancing intelligent environmental management.

6. Practical applications

The hybrid-optimized LSTM model proposed in this study demonstrates significant advantages in enhancing the prediction accuracy and robustness of DO levels, offering a novel technological approach for intelligent water quality monitoring and water resource management. This method is not only suitable for dynamic water quality modeling in diverse environments, such as river basins, lakes, and urban water bodies, but also exhibits strong versatility and scalability, enabling adaptation to environmental data characteristics across different regions and pollution backgrounds. When integrated with Internet of Things sensor networks, the model shows promise for incorporation into real-time water quality monitoring systems. It can facilitate rapid detection and predictive early warning of key water quality indicators, thereby providing technical support for emergency responses to water pollution incidents and for the optimization of treatment strategies.

The precise DO prediction results can further guide the optimization of specific water treatment interventions, such as adjusting aeration intensity, applying chemical oxidants, or implementing bioaugmentation measures. This optimization can improve treatment efficiency and enhance system energy efficiency in managing metals, inorganic pollutants, and organic contaminants. As data-driven approaches become increasingly prevalent in environmental engineering, this hybrid optimization framework can also be extended to the modeling and predicting of other key pollutants in aquatic systems. It thus provides an algorithmic foundation and decision-making support for developing efficient, intelligent, and sustainable pollution control and ecological protection systems.

7. Conclusion

This research presents an innovative hybrid optimization approach that integrates the Nadam optimizer with the DE algorithm. For the first time, this combination is applied to the training of DO prediction models, effectively enhancing the accuracy and stability of deep learning models in water quality time-series modeling. Through the dynamic coordination of local and global search strategies, this method overcame the issue of local convergence common in traditional approaches to nonlinear modeling and demonstrates excellent performance in analyzing highly sequential environmental data.

Compared with existing studies, this study makes two methodological breakthroughs. First, it introduces, for the first time in the field of water quality prediction, a dual-optimizer collaborative framework based on an information interaction mechanism, thus addressing the current lack of engineering applications of intelligent optimization in environmental prediction. Second, it validates the robustness and applicability of this method in water body prediction tasks, highlighting its strong algorithmic novelty and adaptability.

The social contributions of this research are reflected in three key areas. First, the proposed prediction model contributes to the development of a more efficient early warning system for water environments, improving the proactive monitoring and response capacity for pollution events. Second, the technical framework provides data support and modeling foundation for intelligent water resource allocation and governance decision-making. Third, in the context of increasing global water quality risks, it offers a practical solution for constructing sustainable and intelligent systems for water environment management.

From a research significance perspective, this study extends the application of artificial intelligence optimization algorithms in environmental science, strengthening the link between algorithm development and practical needs in environmental engineering. It provides both a methodological basis and a practical direction for future interdisciplinary research.

Despite its promising outcomes, this research has certain limitations. Oscillation phenomena was observed during the local iteration phase of model training. In addition, extensive verification under cross-regional, multi-pollutant, and multi- objective scenarios has not yet been conducted. Future research will aim to enhance training stability and explore the integration of this approach with the Internet of Things sensing systems to establish an end-to-end, closed-loop framework for water quality monitoring and decision-making feedback.

In summary, this paper not only proposes an original hybrid optimization framework at the methodological level but also explores valuable engineering applications of environmental artificial intelligence. It thus holds significant scientific value, engineering significance, and social influence.

Acknowledgments

The authors would like to thank Prof. Azman and Dr. Suhaili for their valuable guidance and suggestions during this research. We also acknowledge the use of open-access datasets.

Funding

None.

Conflict of interest

The authors declare no conflicts of interest.

Author contributions

Conceptualization: All authors
Data curation: Tu Jun
Formal analysis: Tu Jun
Writing – original draft: Tu Jun
Writing – review & editing: Azman Yasin, Nur Suhaili Mansor

Availability of data

Data corresponding to this study are available from the corresponding author upon reasonable request.

References

1. Yang J, Chen TS, Sun Q, Huang BY, Ren SJ. The impact of Urban development upon the quality of drinking water sources: Evidence from China. *IOP Conf Ser Earth Environ Sci*. 2020;508(1):e012072. doi: 10.1088/1755-1315/508/1/012072
2. Liu Z, Ying J, He C, *et al*. Scarcity and quality risks for future global Urban water supply. *Landsc Ecol*. 2024;39(10):1-17. doi: 10.1007/s10980-024-01832-0
3. Ali B, Anuska, Mishra A. Effects of dissolved oxygen concentration on freshwater fish: A review. *Int J Fish Aquat Stud*. 2022;10(4):113-127. doi: 10.22271/fish.2022.v10.i4b.2693
4. Liu W, Lin S, Li X, *et al*. Analysis of dissolved oxygen influencing factors and concentration prediction using input variable selection technique: A hybrid machine learning approach. *J Environ Manage*. 2024;357:120777. doi: 10.1016/j.jenvman.2024.120777
5. Hu Y, Liu C, Wollheim WM. Prediction of riverine daily minimum dissolved oxygen concentrations using hybrid deep learning and routine hydrometeorological data. *Sci Total Environ*. 2024;918:170383. doi: 10.1016/j.scitotenv.2024.170383
6. Doroudi S, Kheyruri Y, Sharafati A, Hameed AS. Enhancing prediction of dissolved oxygen over Santa Margarita River: Long short-term memory incorporated with multi-objective observer-teacher-learner optimization. *J Water Process Eng*. 2025;70:106969. doi: 10.1016/j.jwpe.2025.106969
7. Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M. A survey on long short-term memory networks for time series prediction. *Procedia CIRP.* 2021;99(2):650-655. doi: 10.1016/j.procir.2021.03.088
8. Hikmat Haji S, Mohsin Abdulazeez A. Comparison of optimization techniques based on gradient descent algorithm: A review. *J Archaeol Egypt*. 2021;18(4):2715-2743.
9. Dozat T. Incorporating nesterov momentum into Adam. *ICLR Work*. 2016;1:1-4.
10. Yu Y, Gao S, Wang Y, Todo Y. Global optimum-based search differential evolution. *IEEE/CAA J Autom Sin*. 2019;6(2):379-394. doi: 10.1109/JAS.2019.1911378
11. Ren X, Xu Z, Chu W, *et al*. The discharge of chlorinated effluent from wastewater treatment plants enhances dissolved oxygen in the receiving river: From laboratory study to practical application. *Water Res*. 2025;273(9):123012. doi: 10.1016/j.watres.2024.123012
12. Jiang Y, Zhang L, Wang C, *et al*. Multi-step prediction of dissolved oxygen in fish pond aquaculture using feature reconstruction-based deep neural network. *Comput Electron Agric*. 2025;232(14):109997. doi: 10.1016/j.compag.2025.109997
13. Ji X, Shang X, Dahlgren RA, Zhang M. Prediction of dissolved oxygen concentration in hypoxic river systems using support vector machine: A case study of Wen-Rui Tang River, China. *Environ Sci Pollut Res*. 2017;24(19):16062-16076. doi: 10.1007/s11356-017-9243-7
14. Khan UT, Valeo C. A new fuzzy linear regression approach for dissolved oxygen prediction. *Hydrol Sci J*. 2015;60(6):1096-1119. doi: 10.1080/02626667.2014.900558
15. Wu J, Li Z, Zhu L, Li G, Niu B, Peng F. Optimized BP neural network for dissolved oxygen prediction. *IFAC-PapersOnLine*. 2018;51(17):596-601. doi: 10.1016/j.ifacol.2018.08.132
16. Malek S, Mosleh MA, Syed SM. Dissolved oxygen prediction using support vector machine. *Int J Comput Inform Syst Control Eng*. 2014;8(1):46-50.
17. Tan W, Zhang J, Liu X, *et al*. dissolved oxygen prediction based on PCA-LSTM. *J Phys Conf Ser*. 2022;2337(1):012012. doi: 10.1088/1742-6596/2337/1/012012
18. Hao Z. Research on prediction algorithm of dissolved oxygen in aquatic products based on improved lstm algorithm. *2024 IEEE 3rd Int Conf Electr Eng Big Data Algorithms(EEBDA)*. 2024:86-90. doi: 10.1109/EEBDA60612.2024.10485663
19. Gandh DR, Harigovindan VP, Rasheed Abdul Haq KP, Bhide A. Attention-driven LSTM and GRU deep learning techniques for precise water quality prediction in smart aquaculture. *Aquac Int*. 2024;32(6):8455-8478. doi: 10.1007/s10499-024-01574-5
20. Lee SY, Jeong DY, Choi J, Jo SK, Park DH, Kim JG. LSTM model to predict missing data of dissolved oxygen in land-based aquaculture farm. *ETRI J*. 2024;46(6):1047-1060. doi: 10.4218/etrij.2023-0337
21. Shuai M, Tian H. Long short term memory based on differential evolution in passenger flow forecasting. J Simul. 2019;7(1):17-21.
22. Peng L, Liu S, Liu R, Wang L. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*. 2018;162:1301-1314. doi: 10.1016/j.energy.2018.05.052
23. Storn RM, Price K. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Australas Plant Pathol*. 2009;38(3):284-287.
24. Deng W, Shang S, Cai X, Zhao H, Song Y, Xu J. An improved differential evolution algorithm and its

application in optimization problem. *Soft Comput*. 2021;25:5277-5298.
doi: 10.1007/s00500-020-05527-x

25. Nguyen QV, Miller N, Arness D, Huang W, Huang ML, Simoff S. Evaluation on interactive visualization data with scatterplots. *Vis Informatics*. 2020;4(4):1-10.
doi: 10.1016/j.visinf.2020.09.004
26. Lu Z. Comparison of stock price prediction models for linear models, random forest and LSTM. *Appl Comput Eng*. 2024;54(1):226-233.
doi: 10.54254/2755-2721/54/20241598
27. Gürsoy Mİ, Alkan A. Investigation of diabetes data with permutation feature importance based deep learning methods. *Karadeniz Fen Bilim Derg*. 2022;12(2):916-930.
doi: 10.31466/kfbd.1174591
28. De Smedt T, Daelemans W. Pattern for python. *J Mach Learn Res*. 2012;13(1):2063-2067.
29. Chicho BT, Sallow AB. A comprehensive survey of deep learning models based on keras framework. *J Soft Comput Data Min*. 2021;2(2):49-62.
doi: 10.30880/jscdm.2021.02.02.005
30. Karim F, Majumdar S, Darabi H. Insights into lstm fully convolutional networks for time series classification. *IEEE Access*. 2019;7:67718-67725.
doi: 10.1109/ACCESS.2019.2916828
31. Liao L, Li H, Shang W, Ma L. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Trans Softw Eng Methodol*. 2022;31(3):1-40.
doi: 10.1145/3506695
32. Hodson TO, Over TM, Foks SS. Mean squared error, deconstructed. *J Adv Model Earth Syst*. 2021;13(12):e2021MS002681.
doi: 10.1029/2021MS002681