Article

Sequential GP-UCB Bayesian optimization for deep neural network fine-tuning in dissolved oxygen prediction

Farid Hassanbaki Garabaghi[1], **Semra Benzer**[2], **Recep Benzer**[3]

[1]Department of Environmental Sciences, Graduate School of Natural and Applied Sciences, Gazi University 06500, Ankara, Turkey

[2]Department of Science Education, Gazi Faculty of Education, Gazi University 06500, Ankara, Turkey

[3]Department of Management Information Systems, School of Administrative and Social Sciences, Ankara Medipol University 06050, Ankara, Turkey

E-mail: sbenzer@gazi.edu.tr, sbenzer@gmail.com

Received 29 March 2025; *Accepted 5 May 2025*; *Published online 15 May 2025*; *Published 1 December 2025*

Abstract

Dissolved Oxygen (DO) serves as a crucial measure of water quality, imperative for both aquatic life and human consumption. The application of deep learning, particularly through data-driven predictions, offers a robust tool for estimating DO concentrations. Enhanced precision is achieved by fine-tuning hyperparameters. Bayesian optimization methods are amongst those noteworthy for their effectiveness. This study focuses on predicting DO levels using a Deep Neural Network model. The study uses Bayesian optimization to refine hyperparameters for the best model setup, comparing the results with a baseline model using default settings. Results indicate that the Bayesian-optimized model outperforms the baseline. The findings underscore the pivotal role of Bayesian optimization in elevating model performance, exhibiting robust generalization capabilities while significantly reducing the need for manual parameter tuning. This successful application underscores a substantial methodological advancement in environmental management, particularly in predictive modelling for indicators of aquatic ecosystem health.

Keywords water quality; dissolved oxygen; environmental management; deep learning; Bayesian optimisation; gaussian process.

Computational Ecology and Software
ISSN 2220-721X
URL: http://www.iaees.org/publications/journals/ces/online-version.asp
RSS: http://www.iaees.org/publications/journals/ces/rss.xml
E-mail: ces@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1Introduction

One of the critical water quality indicators, Dissolved Oxygen (DO), serves as a measure of aquatic ecosystem health (Rouf et al., 2022). A healthy waterbody with ample concentration of DO not only supports the ecosystem but also fulfills human requirements for drinking, irrigation, and recreation (Tiyasha et al., 2021). It is a well-established fact that minor variations in DO concentration can significantly affect aquatic life (Fitri et

al., 2021). Furthermore, slight fluctuations in other water quality variables can dramatically alter DO concentrations (Garabaghi et al., 2023). Due to the sensitivity of DO to any changes in a waterbody monitoring DO is often prioritized as the primary parameter for providing a comprehensive overview of a water system's health (Kannel et al., 2007).

Among various modelling techniques designed to capture the temporal fluctuations of DO levels, machine learning methods are suggested as effective solutions for addressing the challenges posed by nonlinearity (Ziyad Sami et al., 2022). Machines are computers that learn from their experiences and make sense of the world using a structured set of concepts. Each concept builds upon simpler ones. This method circumvents the need for humans to input all the knowledge a computer requires (Zhang et al., 2008; Agahian and Akan, 2022). The computer gradually learns more complex ideas from the basic ones. The structure of all simple concepts upon each other is complex and multi-layered, that is called deep learning (LeCun et al., 2015; Goodfellow et al., 2016).

Deep learning models have been in the centre of focus of many researchers aimed to propose predictive models for DO concentration. Banerjee et al. (2019) presented a study in which artificial neural network modelling was used to predict dissolved oxygen levels. It compared the efficacy of deep learning models against traditional regression in correlating environmental factors with aquatic ecosystem health indicators. The study concluded that neural network models are better at predicting these variables, which can significantly aid in the management of water resources. Moghadam et al. (2021) benchmarked a Recurrent Neural Network (RNN) in predicting DO levels against support vector machine (SVM) and artificial neural network (ANN) models. Results showed the RNN model outperformed others, with high accuracy in DO prediction for different time leads, emphasizing the potential of deep learning in environmental monitoring. However, recurrent neural networks are considered computationally expensive models due to the sequential nature of their computations, and backpropagation through time process for training the network (Pascanu et al., 2013; Sutskever et al., 2014; Goodfellow et al., 2016; Sherstinsky, 2020). Zhi et al. (2021) investigated whether a deep learning model, specifically, a Long Short-Term Memory model (LSTM), can accurately predict DO levels across watersheds. The key finding of this study is that the model struggles to predict extreme peaks and troughs in DO concentrations. Even more interestingly, they reported that the model's performance did not improve with more data. The findings suggested that more targeted data collection is necessary to improve predictions. Zhu et al. (2021), explored the use of deep learning and transfer learning to predict DO concentrations in aquatic systems. They developed a pre-trained model using a large dataset from one aquatic system and applied it to another system with less data. They compared the model's performance with and without transfer learning and found that transfer learning improved the prediction of DO concentrations. Their study demonstrated the potential of using transfer learning to predict environmental parameters in different aquatic systems with limited data.

It is an ongoing discussion that deep learning models can perform tasks with varying dataset sizes. However, the quality of predictions and the reduction in false positives improve with larger datasets. Although, deep learning models can function with smaller datasets, the dimensionality and breadth of data can significantly impact the model's refinement and accuracy (Ionescu et al., 2023). Another task that poses a challenge in developing deep net models is fine tuning the hyperparameters. It takes a deep expertise, and trial and error to find the best hyperparameters to optimise a deep learning algorithm (Xiao et al., 2020). However, there are many methods for hyperparameter optimisation such as hyperparameter optimization via radial basis function (RBF) and dynamic coordinate search (Ilievski et al., 2017), genetic algorithms (Zhang, 2016; Thi Kieu Tran et al., 2020; Xiao et al., 2020), reinforcement learning (Rijsdijk et al., 2021), etc. Amongst all the methods for fine tuning deep nets, Bayesian approach has attracted the attention of many researchers so far.

Bayesian methods suggest to solve problems associated with uncertainty (Sun, 2013). This approach integrates existing knowledge about an unknown function with data collected from samples. Using the Bayesian formula, it derives the function's distribution as posterior information. From this updated information, the optimal value of the function can then be inferred (Zhang, 2010, 2016; Wu et al., 2019). Then, Bayesian methods have been used in variety of research domains where uncertainty was challenge, and it has been used in developing machine learning models to address the challenge associated with uncertainty of hyperparameter selection (Huan et al., 2018; Chandra and Tiwari, 2022; Joy et al., 2016a; Yan et al., 2023).

One of the debates about using Bayesian optimisers in machine learning models associates with its complexity of computation, especially where high-dimensional problem must be addressed (Peter I. Frazier, 2018).

As discussed previously, deep learning models can predict DO values with a high degree of accuracy. However, their performance is often contingent on large datasets to ensure robust correlation and generalization. This study tackles the challenge of enhancing model accuracy using datasets of moderate size. A pivotal aspect is the optimization of hyperparameters, which, while effective in improving model performance, often entails a time-intensive trial-and-error process. Bayesian optimization presents a promising alternative, evidenced by its success across various domains. Despite its computational intensity, integrating this method with deep learning models could be transformative.

To address these challenges, we introduce a deep learning model that leverages a computationally efficient Bayesian optimizer. This study evaluates the viability of Bayesian optimization for hyperparameter tuning within a deep learning context, investigating whether the computational investment in such a tool is justified by the resultant improvements in model generalization and performance. Our contribution lies in developing a model that stands at the intersection of computational feasibility and environmental management, potentially revolutionizing the prediction of DO levels in riverine systems.

2 Methods

2.1 Study area

Büyük Menderes basin locates in Turkey's southwest. The basin’s longest river, Büyük Menderes river, stretches 584 km from the high eastern mountains. The basin covers 3.2% of Turkey's area, and it ranks as the seventh most densely populated basin, with an annual average flow, precipitation, discharge, and temperature at 3,020 billion cubic meters, 635 mm, 110 cubic meters per second, and 17.6 °C, respectively (EUP, 2016; Akyildiz and Duran, 2021).

The water quality data used in this study was collected from 8 stations along the Büyük Menderes River. These stations represent the main flow of the basin (Fig. 1).

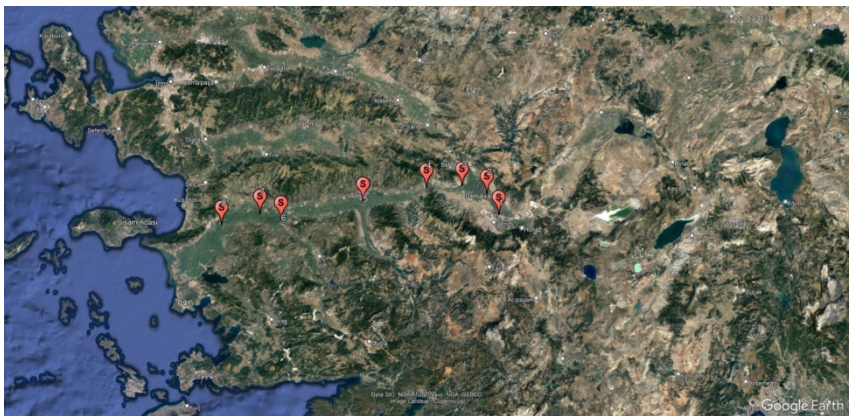


Fig. 1 Study area water quality monitoring stations.

The dataset comprises water quality measurements taken every two months between 2004 and 2014. Using the method outlined by Kumar and Manjula (2012), a homogenous dataset consisting of 528 instances and 19 parameters, including year, month, station, temperature, pH, electrical conductivity, chemical oxygen demand, biological oxygen demand, ammonium nitrogen, nitrite, nitrate, phosphorous, Chloride, Iron, manganese, sodium, sulphate, total dissolved solids, and dissolved oxygen, was created.

2.2 Bayes' theorem and Bayesian optimisation

Fine-tuning of hyperparameters in a machine learning model is a significant challenge as the correct set of hyperparameters can boost up performance of a model (Probst et al., 2019; Weerts et al., 2020). Bayesian optimisation is an efficient method for tuning settings or configurations when calculating or finding the best setting is a challenge (Snoek et al., 2012). The nature of Bayesian optimisation is built upon Bayes' Theorem. Bayes Theorem is used to update the probability estimate for a hypothesis as more evidence and information becomes available (Zhang, 2010). The theorem is usually written as Eq. 1, where $p(\theta)$ is prior probability that is what we initially believe about the distribution of parameter θ. In this equation $p(y \mid \theta)$ is the likelihood function that gives information how likely the observed data y is given the parameter θ, and finally posterior distribution as $p(\theta \mid y)$ is computed that is the updated belief about the distribution of θ after seeing the data y (Zhang, 2010; Box and Tiao, 2011).

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)} \quad (1)$$

Based upon Bayes' Theorem, Bayesian optimisation simplifies the best hyperparameter combination exploration as there are numerous of possible combinations whose exploration by trial and error is time-consuming (Joy et al., 2016). What is updated in Bayesian optimisation is the surrogate model that is the prior belief about the unknown function space. As more observations are gathered, these models are updated according to Bayes' theorem, incorporating the new data to improve the model's accuracy and the subsequent selection of points for function evaluation (Snoek et al., 2012). The standard objective of Bayesian optimisation is to diminish the sum of Regret, $\mathcal{R}$T, as Eq. 2, where $f(x^*)$ is the true optimum of f (f: X →ℝ) over a set X (X ⊂ℝ d), the unknown function that is a subset of multidimensional space ℝ d denoted as Eq. 3. In other word, the goal is to find the maximum value of the unknown function (Kandasamy et al., 2018).

$$\sum_{t=1}^{T}(f(x^*) - f(x_t)) \quad (2)$$

$$f(x^*) = \max f(x), x \in X \quad (3)$$

In a sequential Bayesian optimisation, a batch of $\mathcal{K}$ points at each iteration, t, from the space X is chosen. These points are called queries and denoted as $\{x_t^k\}\ 0 \leq k < K$. Here, for each point, the actual function f is not directly observed but a noisy version of the function's value is observed (Eq. 4) (Srinivas et al., 2009).

$$y_t^k = f(x_t^k) + \epsilon_t^k \quad (4)$$

where, ϵ_t^k is Gaussian noise $\mathcal{N}$ with mean 0 and standard deviation of σ^2.

2.3 Gaussian process and upper confidence bound (GP-UCB)

Gaussian process is a statistical method that updates the intuition on prior distribution (function) by assuming that the function's values at different points are random variables that have a joint Gaussian distribution (Liu et

al., 2023). In other words, it means that unknown function f is a sample from GP. So, that means if any points is taken from f, the values at these points would collectively follow a multivariate Gaussian distribution. Let's say that the unknown function f has a mean function m and a non-negative definite kernel function k (Contal et al., 2013):

$$f \sim \text{GP}(m, k), \quad (5)$$

where m: X →ℝ, and k: X × X → $\mathbb{R}^+$

In other words, m is the way to describe the prior belief about the function's behaviour as m(x) is the best guess for value f at any point x. the kernel or covariance function k gives us information on expectation from covary of the function's values between two points x and x'.

When a selection of points taken from an unknown function, and the outcomes at these points jointly follow a Gaussian distribution, this distribution is characterised by a mean vector μ and a covariance matrix C. These elements are derived from the process's mean function and its covariance function (kernel) within the GP framework. This holds true for any number of points n taken from the process's domain $x_1, \dots, x_n \in X$ (Contal et al., 2013; Joy et al., 2016).

$$\big(f(x_1), \dots, f(x_n)\big) \sim \mathcal{N}(\mu, C), \quad (6)$$
$$\text{with} \mu[x_i] = m(x_i)$$
$$\text{and} C\big[x_i,\ x_j\big] = \ k(x_i,\ x_j)$$

When we gather data from the function f, we can update our GP model. This is where Bayesian inference comes in. The posterior distribution then can be calculated for the function's values at new points by combining our prior model that is the GP with the data. This posterior GP gives us a new mean and variance for each point, which represents our updated belief about the function's values after considering the data. The updated mean (Eq. 7) is our revised best guess for f(x), and the updated variance (Eq. 8) quantifies our uncertainty (Rasmussen and Williams, 2005).

$$\hat{\mu}_{T+1}(x) = \ k_T x^T C_T^{-1} Y_T \quad (7)$$
$$\hat{\sigma}^2_{T+1}(x) = k(x,x) - \ k_T(x)^T C_T^{-1} k_T(x) \quad (8)$$

Here, $X_T = \{x_T^k\} t < T, k < K$ is the set of queried locations, $Y_T = \big[y_t^k\big]_{x_t^k \in X_T}$ is the vector of noisy observations, $k_T(x) = \big[k(x_t^k,\ x)\big]_{x_t^k \in X_T}$ is the vector of covariance noisy observations between x and the queried points, and $C_T = K_T + \sigma^2 \text{I}$ is the kernel matrix.

It can be inferred from the equations above that the updated mean $\hat{\mu}_{T+1}(x)$ is calculated using the covariance between x and the observed points, weighted by the inverse of the covariance matrix from the observed data and the observed outputs, and the updated variance $\hat{\sigma}^2_{T+1}(x)$ reflects how much we don't know about f(x) after seeing the data. It's calculated based on the kernel's self-covariance at x minus a correction term that accounts for what the data has taught us (Contal et al., 2013).

Within the Gaussian Process (GP) model, a fundamental aspect is that the posterior distribution at a specific point is normally distributed. This allows for the creation of an upper and lower confidence bound (Eq. 9 and 10) around the estimated mean of the GP at that point.

$$\hat{f}_T^+ = \hat{\mu}_T(x) + \sqrt{\beta_T}\hat{\sigma}_T(x) \quad (9)$$

$$\hat{f}_T^- = \hat{\mu}_T(x) - \sqrt{\beta_T}\hat{\sigma}_T(x) \quad (10)$$

where, $\beta_T \in \mathcal{O}(logT)$. These bounds are set in such a way that the true value of the function is likely to fall within them. The breadth of these bounds is determined by a scaling factor that adjusts with the number of iterations, aiming to ensure a high level of confidence in the interval (Contal et al., 2013).

2.4 Model and workflow

In our research, we have implemented a sequential deep learning model consisting of Three main layers. The hidden layer is a densely connected layer employing the Rectified Linear Unit (ReLU) activation function. This choice is motivated by ReLU's ability to introduce non-linearity into the model while mitigating the vanishing gradient problem commonly encountered in deep networks (Glorot et al., 2011). The output layer is a dense layer with a single neuron utilizing a linear activation function. This configuration is standard for regression tasks where the goal is to predict a continuous outcome (Goodfellow et al., 2016). For training the network, we have selected the Adam optimizer. This optimizer is renowned for its computational efficiency and robustness, featuring adaptive learning rate adjustments, which are particularly advantageous for handling the sparse and noisy gradients that often arise in complex optimization landscapes (Kingma and Ba, 2014). Fig. 2 illustrates the schematic workflow of this study.

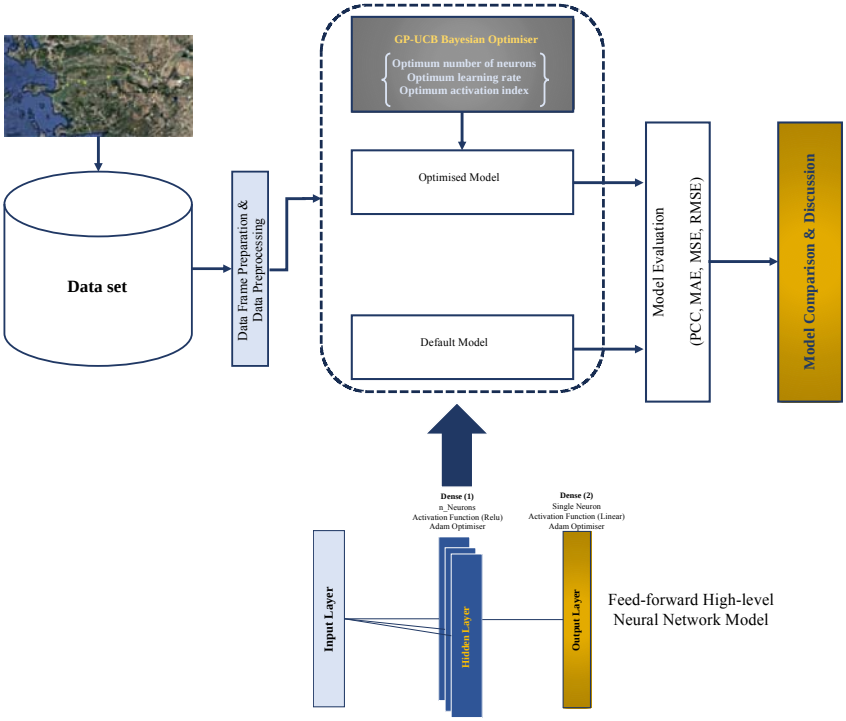


Fig. 2 Schematic workflow of the model development and evaluation.

The data were collected and provided by the General Directorate of State Hydraulic Works in Türkiye. A data integration process was undertaken to compile a cohesive dataset, thoughtfully designed to align with the objectives of the study. Subsequently, the data were cleaned and scaled to minimize noise as effectively as possible. To determine the optimal hyperparameters for the model, we employed a Gaussian Process Upper

Confidence Bound (GP-UCB) Bayesian Optimizer. This optimizer was tasked with identifying the ideal number of neurons, learning rate, and activation function—a parameter we refer to as the activation index—to enhance the model's performance. The model was then trained using the hyperparameters optimized by the Bayesian approach. The performance of this Bayesian-optimized model was evaluated using the metrics described in the previous section and benchmarked against a model configured with default hyperparameters.

2.5 Evaluation metrics

Four evaluation metrics were employed to assess the performance of the models. The Mean Absolute Error (MAE) quantifies the average magnitude of the absolute errors between the observed and predicted values, as defined in Eq. 11. The Root Mean Squared Error (RMSE) is a metric based on squared differences, outlined in Eq. 12. The RMSE provides a scale-sensitive error measure by computing the square root of the Mean Squared Error, so it reverts the measurement to the same scale as the original data. The reason for choosing these metrics for evaluation of the models is their focus on the average sparseness without considering bias (Steurer et al., 2021). In this study, the Pearson correlation coefficient (PCC) was utilized (Eq. 13) to assess the robustness and the linear relationship between the actual and predicted values (Li et al., 2022).

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|p_n - \hat{p}_n| \quad (11)$$

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(p_n - \hat{p}_n)^2}{N}} \quad (12)$$

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} - \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (13)$$

3 Results

3.1 Convergence analysis of the optimiser

The convergence plot of the Bayesian optimiser used in this study is shown in Fig. 3 representing the performance of the optimiser over iterations. The target score used for evaluation of the performance of the optimiser is negative MSE. The reason for using the negative value of MSE is for the fact that Bayesian optimiser tend to maximise the target function (Snoek et al., 2012; Gabler and Wollherr, 2022). In other words, minimising the MSE results in proceeding the optimisation in a correct way so adopting the negative amount of MSE allows us to reach this goal. Therefore, in this Fig. 3, the lower the values of MSE, the better model performance, and vice versa. The initial phase of the plot is characterised by significant variations of target score with improvements and deteriorations between continuous iterations. Here, the optimiser undertakes the exploration. In this phase, the optimizer is sampling the parameter space to identify regions of high performance. Although spaces of high performance can be identified in between the fluctuations, the overall trend of this phase in downward. This suggests that over time, the optimizer is successfully finding hyperparameters that lead to better model performance (lower MSE). The plot indicates that up to around iteration 40, the models starts to converge. Although there is a significant fluctuation after iteration 40, the values of MSE are less negative compared to the initial values of MSE. This suggests that the optimizer has likely identified a region of the hyperparameter space that yields relatively good model performance and is fine-tuning within that region. When it comes to the stability of the convergence, the plot indicates that the stability is not achieved completely as the target scores have not fixed around a single value. This may suggest that the optimiser avails of several factors such as 1) more iterations, 2) different selection of hyperparameters, 3) change in the acquisition function, and 4) adjustment to the exploration-exploitation (Bull, 2011; Orlando

Romero et al., 2020). In summary, the plot shows that the Bayesian optimiser improves the model's performance over iterations, with a trend towards better hyperparameter configurations as indicated by less negative MSE scores.

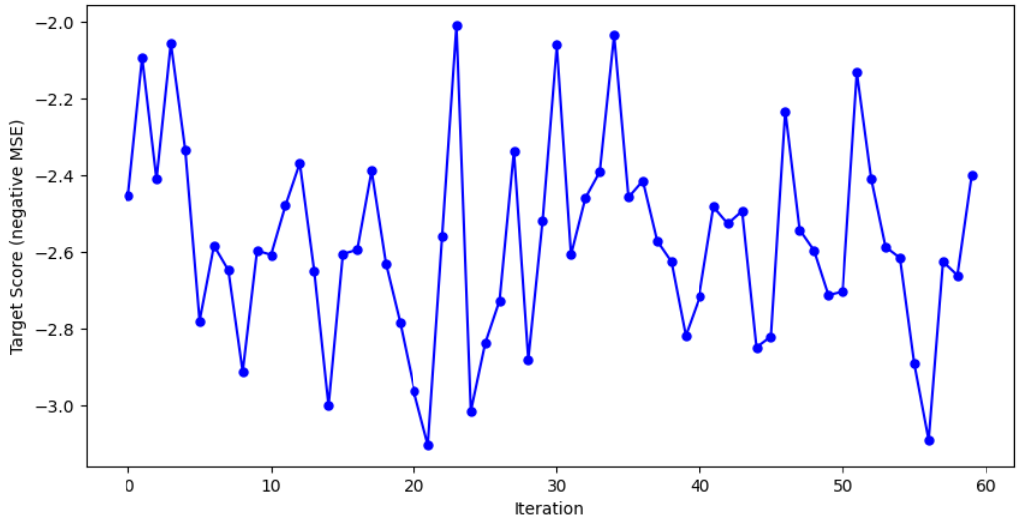
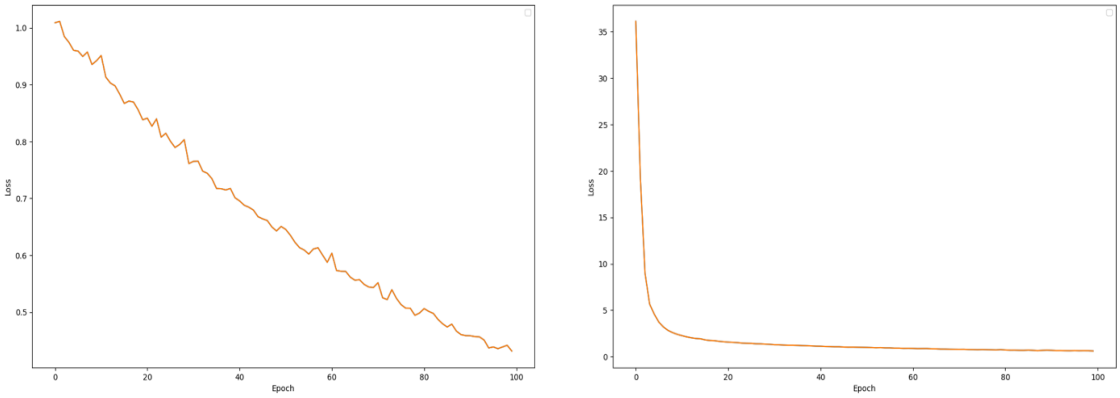


Fig. 3 Convergence plot of the Bayesian optimiser.

3.2 Training and validation analysis

Fig. 4 illustrates the models loss during validation phase. The loss plot of the Bayesian optimised model (Fig. 4a) shows a consistent descend in 100 epochs. This indicates that the model learns from training data without running into any overfitting problem as the plot starts at a loss of approximately 1.0 and decreases to around 0.1 loss without any upward trend. As there is no sharp drops in the plot, it can be concluded that the learning process is stable. This smoothness also suggests the positive effect of the updated hyperparameters on the models learning.

On the other hand, Fig. 4b indicates the loss plot of the default model. This plot indicates a steep decline in loss during the initial epochs, then it reaches to a plateau for the entire epochs. The plot indicates that the loss drops down dramatically from a high value of roughly 35 to somewhere around zero. This suggests that the models learns quickly during initial epochs. However, quickly reaching a flat line indicates that the model may not be improving significantly after the initial learning phase. That may be for the fact that the model reaches its capacity of learning. General downward trend of the plot suggest that there is no overfitting problem. This is considered as a good indicator for a stable learning.



(a) **(b)**

Fig. 4 Loss plots of the validation: a) Loss plot of the Bayesian optimised model, b) Loss plot of the default model.

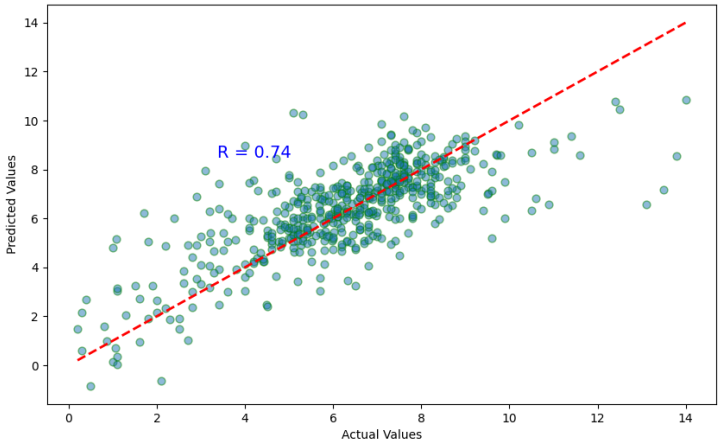
3.3 Models' performance analysis

In this study, PCC, MAE, and MSE are considered as metrics to evaluate the performance of the models as shown in Table 1. Here, the higher PCC value of the Bayesian optimised model (0.955) compared to the PCC value of default model (0.74) suggest a stronger linear relationship between the actual and predicted values. Besides, Bayesian optimised model has a lower MAE value (0.454) compared to the default model's MAE (1.049). This demonstrates that the Bayesian optimised model's predictions are closer to the actual values on average. It is supported by the results of RMSE of the Bayesian optimised (0.643) and the default model (1.447) as the lower value of the RMSE of the Bayesian optimised model indicates more accurate predictions from the model.

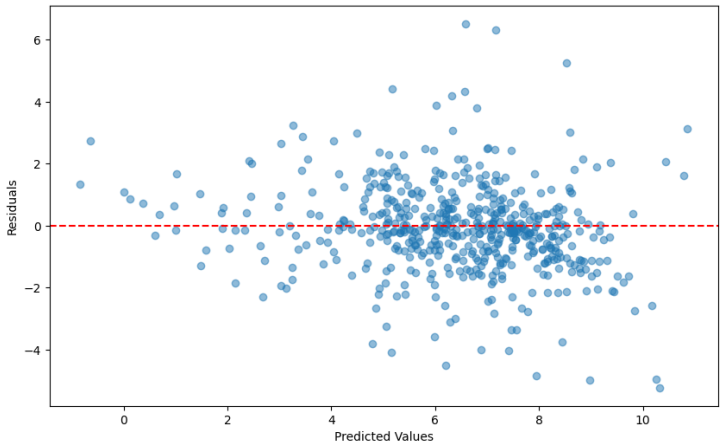
Table 1 Performance evaluation metrics of the models.

	PCC	MAE	RMSE
Default Model	0.74	1.049	1.447
Bayesian-Optimised Model	0.955	0.454	0.643

Fig. 5.a is the scatter plot of the actual against predicted values of DO for the default deep neural network model, that shows the relationship between actual and predicted values. This plot along with the PCC value of 0.74 indicate a moderate positive linear relationship. The sparsity around the best-fit line indicates variability in the accuracy of the predictions. Fig. 5.b shows the residual plot of the default model, that is the differences between actual and predicted values plotted against the predicted values. An ideal residual plot is characterised by a random dispersity of the points around the horizontal axis at zero without any distinguished pattern. Fig. 5.b indicates that residuals are centred around zero suggesting no bias, however, the noticeable sparsity indicates variability in prediction accuracy.



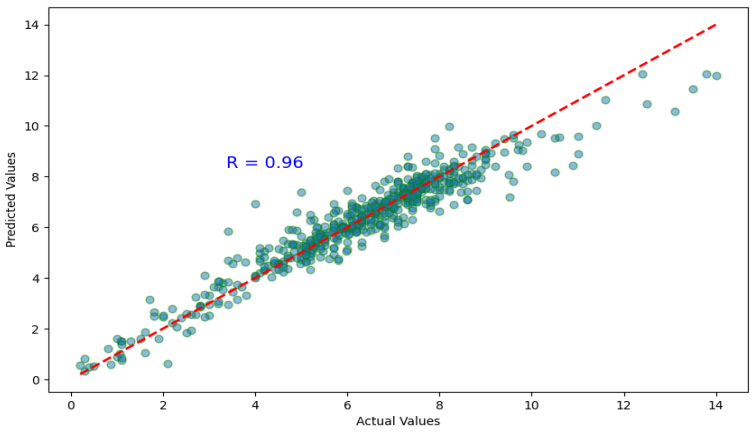
(a)



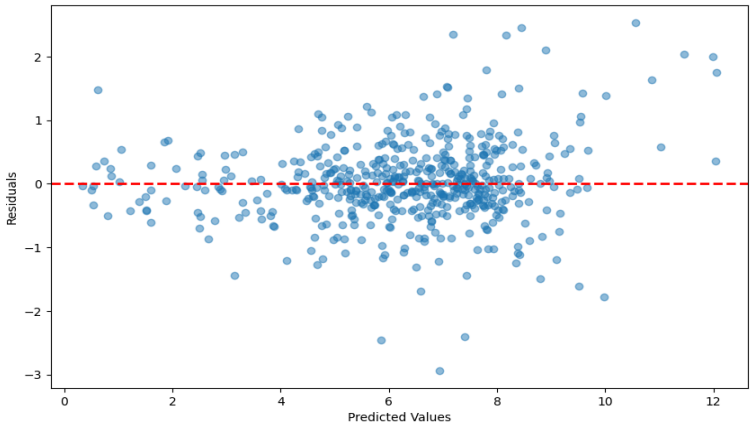
(b)

Fig. 5 Default deep neural network model's performance plots: a) Scatter plot of the actual vs predicted, b) Residual plot of the predictions.

Fig. 6a shows the scatter plot of the actual against predicted values of DO for the default deep neural network model. High value of PCC (0.96) suggests an excellent linear relationship between the actual and predicted values. The tight cluster of data points around the best-fit line indicates that the Bayesian optimised model's prediction are extremely close to the actual values. In Fig. 6b that illustrates the residual plot of the Bayesian optimised model, although there is some sparsity, there is no clear pattern or systematic deviation from the red dashed line at zero. This suggests that the model has no obvious biases in prediction across the range of values.



(a)



(b)

Fig. 6 Bayesian optimised deep neural network model's performance plots: a) Scatter plot of the actual vs predicted, b) Residual plot of the predictions.

In summary of the above analysis, Bayesian optimised deep neural network model outperformed the deep neural network model with default hyperparameters in terms of both linear correlation and residual distribution. The higher PCC value along with the random sparsity of the residuals suggest that the Bayesian Optimized model has a better fit to the data so better generalisation. The Bayesian optimized model's predictions does not only show great intimacy to the actual values but also indicates consistency across the range of predictions without systematic deviations.

3.4 Models' accuracy analysis

Fig. 7 is provided to visually represent the distribution of residuals of the default deep neural network model. The distribution of the residuals (kernel density estimate line (KDE)) seems to be symmetrical normal distribution with central tendency around zero and tails extending roughly equally in both directions. This symmetry suggests that the model does not have systematic bias either overfitting or underfitting. It is indicated in the plot that roughly 68% of the residuals fall within the first standard deviation from the mean, and Approximately 95% of the data falls within two standard deviations from the mean that is consistent with the empirical rule. The residuals fall outside the acceptable confidence interval are considered outliers that significantly affect the accuracy of the model. As seen in the plot, the tails of the KDE extending on both sides

to up to standard deviation 6 on the positive side and roughly standard deviation 6 on the negative side. This suggests although the default model's errors are well-behaved and mostly fall within acceptable limits, but the presence of outliers suggests there may be instances where the model's predictions are significantly off.

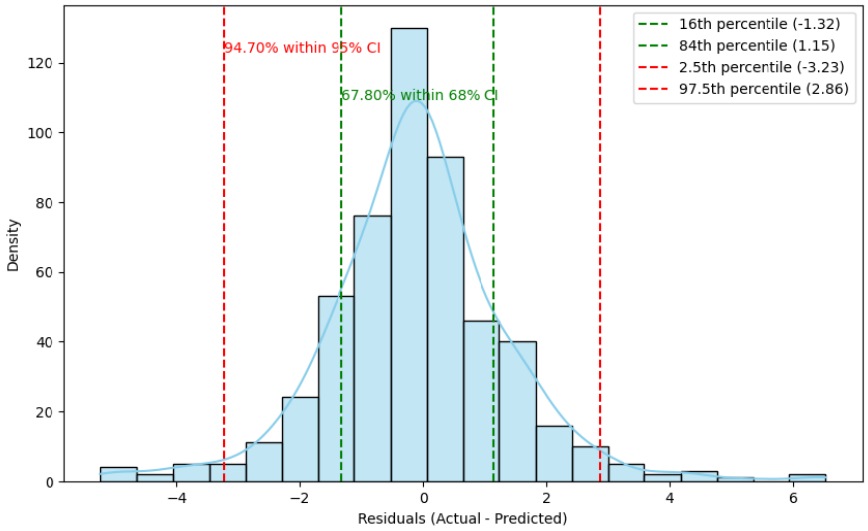


Fig. 7 Kernel density estimate (KDE) plot of the errors of the default deep neural network model.

The violin plot of the actual and predicted values is presented in Fig. 8 to have a comprehensive view of the default deep neural network model's accuracy. As seen in this plot, the central tendency of both actual and predicted distributions is very close with the mean of actual 6.29 and predicted 6.39. This suggest that the model predicts the central tendency of the data well. When it comes to the percentiles, the 25th percentile is slightly higher for predicted (5.37) than for actual (5.20). This is when the 75th percentiles are equal (7.60). This indicates that the predicted values are less sparse in the lower range. However, the sparsity fits well at the upper end. Both actual and predicted values show roughly symmetric distribution around the mean. However, the predicted values slightly skew towards the lower values that suggests that the model may smooth over the lower end of the data distribution.

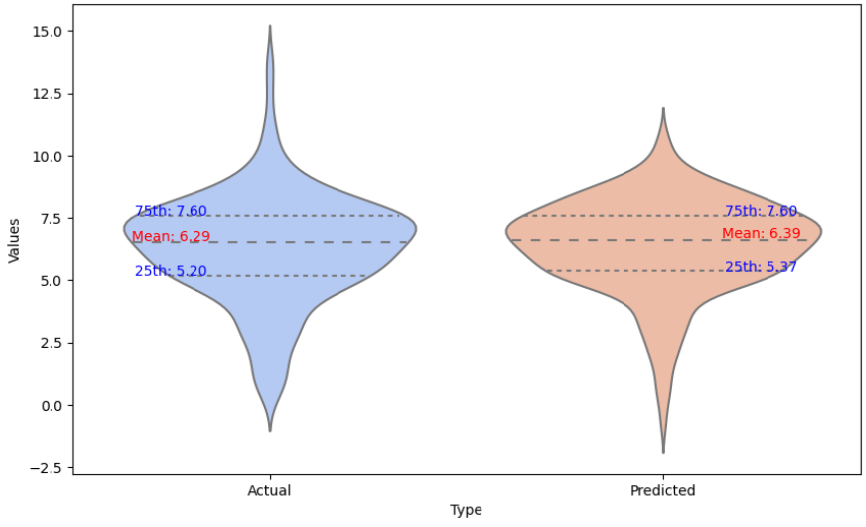


Fig. 8 The violin plot of the actual and predicted values for the default deep neural network model.

In summary, the violin plot supports the KDE plot's analysis. This suggests that the default deep neural network model performs well across the range of values, with errors that are approximately normally distributed and a prediction accuracy that is consistent, as indicated by the central tendency and spread of the data.

Fig. 9 shows the KDE plot of distribution of residuals for the Bayesian optimised deep neural network model. As seen, the pick of the KDE plot is close to zero. It suggests that the model's predictions are very close to the actual values that is an indicative of a perfect model accuracy. The plot is characterised by a very symmetric distribution around the mean with tails extending almost equally on both sides that indicates a perfect balance in overprediction and underprediction. Almost 68% of the residuals locate within one standard deviation, and roughly 95% within two standard deviation. This is in line with the empirical rule for a normal distribution, suggesting the model's predictions are consistent. Catching a glance at the outliers suggest that extreme predictions are rare for the Bayesian optimised model. What else supports the models accuracy is the percentiles (16th = -0.47 , 84th = 0.55, 2.5th = -1.20, and 97.5th = 1.47) that are relatively close to the mean, suggesting a perfect accuracy.

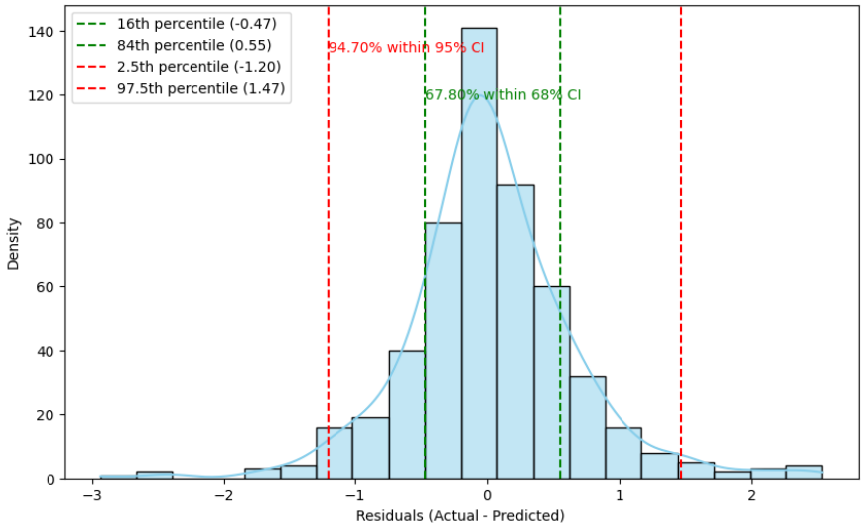


Fig. 9 Kernel density estimate (KDE) plot of the errors of the Bayesian optimised deep neural network model.

Compared to the KDE plot of the default deep neural network model, Bayesian optimized model has a tighter distribution of residuals, indicating more precise predictions. The narrower confidence intervals and the closer proximity of percentiles to the mean in the Bayesian optimized model's KDE plot imply an improvement over the default model.

Fig. 10 shows the violin plot of distribution of actual and predicted values for the Bayesian optimised model. As indicated on the plot, the mean of the actual and predicted values are literally identical ($\mu_{Actual} = 6.29$and $\mu_{Predicted} = 6.27$). This suggest that the model's predictions are extremely accurate on average. Extreme intimacy of the both 25th and 75th percentiles of the actual and predicted distributions as are 5.20 vs. 5.25 for the 25th percentile and 7.60 for both on the 75th percentile, suggest that the model captures the sparsity of the data with outstanding accuracy. As the shapes of the distributions for actual and predicted values are almost identical, the model has learned the data distribution effectively. However, there is a slight discrepancy at the tails that is condonable. Together with the KDE plot of the residuals, violin plot of the

distributions indicate that the Bayesian optimized deep neural network model has achieved a high level of accuracy, with its predictions closely mirroring the actual values across the central tendency and spread.

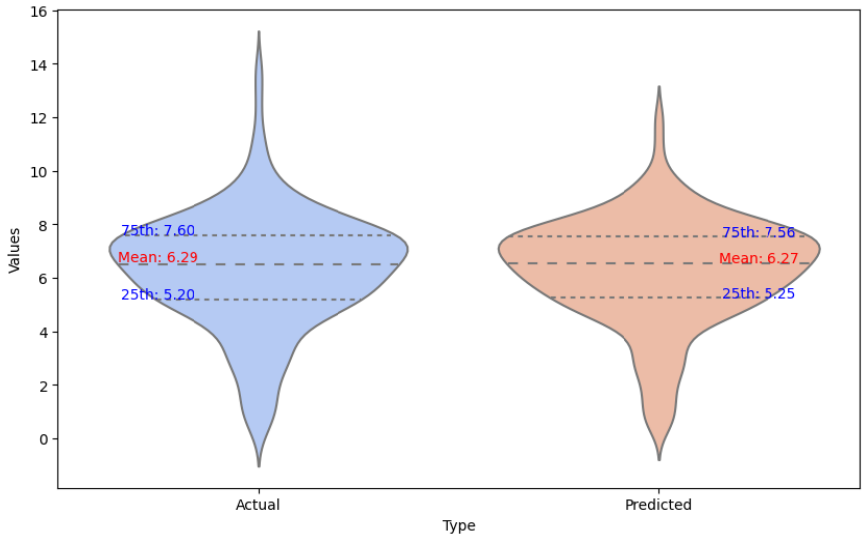


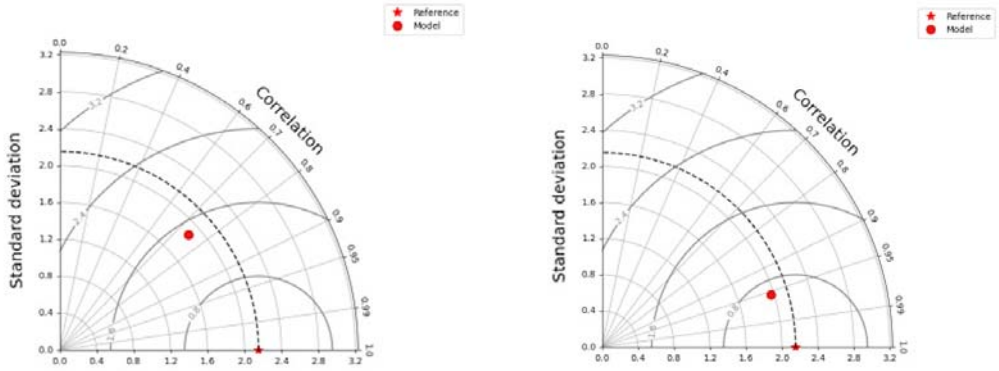
Fig. 10 The violin plot of the actual and predicted values for the Bayesian optimised deep neural network model.

4 Discussion

Effective basin management hinges on data-driven decision-making, underpinned by accurate assessments of water quality. This critical determinant of basin health is intricately linked to complex environmental and human influences (Azma et al., 2023). DO is a crucial water quality parameter that reflects the health of a basin. Rigorous investigation into DO levels is necessary. Accurately predicting its temporal fluctuations can significantly enhance decision-making processes aimed at addressing the complex challenges associated with basin management (Yang, 2023). Deep learning models offer a streamlined and accurate approach for modelling the temporal fluctuations of DO. Yet, despite their promise, they present challenges, particularly in hyperparameter tuning, which adds a layer of complexity and can limit their practical application (Stefan Falkner et al., 2018).

This study explores an approach to enhance the performance of deep neural network (DNN) models through the optimization of hyperparameters, which traditionally presents a challenge due to the vastness of the hyperparameter space. Manual tuning of these hyperparameters is often impractical because it requires navigating a high-dimensional space to identify the optimal configuration (Bischl et al., 2023). In this study, we adopted a Gaussian Process Upper Confidence Bound (GP-UCB) Bayesian optimisation method to explore the optimum configuration of hyperparameters. Then, we benchmarked the deep neural network model whose hyperparameters was adjusted by the Bayesian optimiser against the default deep neural network model. The results highlights the improved performance of the Bayesian-Optimized model over the Default model, as evidenced by the metrics (PCC, MAE, MSE, RMSE), KDE, and violin plots that discussed in previous sections. Evidently, Hoy et al. (2022), developed a Bayesian-optimised artificial neural network to forecast country-scale municipal solid waste (MSW) composition trends. As a result of this study, the Bayesian-optimized ANN models showed smaller relative standard deviations compared to default ANN models, indicating more accurate forecasts. This is in line with the findings of our study and Fig. 11a and b shows how set of patterns matches observations in this study and summarise the findings in one diagram. The

higher standard deviation in the diagram representing the default model (Fig. 11a), indicates that the predictions from this model are more sparse than the observed data. Conversely, the Bayesian-optimized model has a lower standard deviation (Fig. 11b), meaning its predictions are more closely clustered around the mean and better match the observed data's spread. What else justifies the positive impact of hyper parameter tuning in Bayesian-optimised model is the model's closeness to the perfect correlation of 1.0, implying a better linear relationship between the predictions and the observed data compared to the default model.



(a) **(b)**

Fig. 11 Taylor diagram of the models: a) Taylor diagram of the default deep neural network model, b) Taylor diagram of the Bayesian-optimised deep neural network model.

We have explored the optimum configuration space of three hyperparameters in this study (i.e. learning rate, number of neurons, activation index). The importance of selecting the optimum learning rate is due to its capability of converging a model. Too high learning rate may cause overshoot the minimum loss, leading to unstable training and divergence, and too low learning rate may require more epochs for convergence that can be time-consuming and computationally expensive (Bengio, 2012; Goodfellow et al., 2016). Number of neurons is another key elements in deep neural networks affecting models' capacity, generalisation, training time, and overfitting (Bashiri and Farshbaf Geranmayeh, 2011; Bashiri and Farshbaf-Geranmayeh, 2013). Another key factor affecting the performance of neural networks is the activation function that enables the networks to capture non-linear relationships in the data (Johannes Lederer, 2021). The importance of proper selection of these hyperparameters clearly justifies the application of a Bayesian hyperparameter optimiser, and the findings of this study and other exemplary studies in the literature demonstrate the positive impact of tuning the hyperparameters in deep nets. For instance, Parsa et al. (2020), adopted a Bayesian optimiser for designing neural network accelerators efficiently. The results of this study shows the positive impact of multi-objective hyperparameter optimization by Bayesian optimiser on the performance of deep neural network models.

Bayesian optimizers have a positive impact on the generalization performance of neural networks, yet they are not without limitations. A significant drawback is their computational complexity, which leads to resource-intensive models. This complexity stems from the need to evaluate the surrogate model at each iteration, which can be computationally demanding (Lu et al., 2023). Consequently, it raises the question of whether the benefits of employing a costly tool like Bayesian optimizers outweigh the drawbacks, or if alternatives should be considered. Despite these concerns, our study's findings affirm the value of Bayesian

optimizers for hyperparameter tuning. Optimal hyperparameter settings can markedly enhance the performance and accuracy of deep neural network models. Furthermore, Bayesian optimizers eliminate the need for time-consuming and error-prone manual tuning of the hyperparameter space through trial and error.

5 Conclusions

In this study, a DNN model was utilized to predict DO levels in riverine environments. The GP-UCB Bayesian optimization approach wasemployed to fine-tune hyperparameters, seeking the optimal configuration. We benchmarked the optimized model against a baseline model with default settings. The results illustrate that the Bayesian-optimized model significantly surpasses the baseline, especially with moderately sized datasets. Bayesian optimization has thus proven to be instrumental in enhancing model performance, offering strong generalization while greatly reducing manual tuning efforts. This successful application heralds a methodological leap in environmental management, advancing predictive modelling for aquatic ecosystem health indicators.

Acknowledgments

Special thanks are expressed to the General Directorate of State Hydraulic Works, Department of Survey, Planning and Allocations for the provision of the water quality data of Büyük Menderes Basin and all the supports thereafter.

References

Agahian S, Akan T. 2022. Battle royale optimizer for training multi-layer perceptron, Evolving Systems, 13(4): 563-575. https://doi.org/10.1007/s12530-021-09401-5

Akyildiz GK, Duran M. 2021. Evaluation of the impact of heterogeneous environmental pollutants on benthic macroinvertebrates and water quality by long-term monitoring of the Buyuk Menderes River Basin, Environmental Monitoring and Assessment, 193(5): 1-23

Azma A, Liu Y, Azma M, et al. 2023. Hybrid machine learning models for prediction of daily dissolved oxygen, Journal of Water Process Engineering, 54: 103957. https://doi.org/10.1016/j.jwpe.2023.103957

Banerjee A,Chakrabarty M,Rakshit N, et al. 2019. Environmental factors as indicators of dissolved oxygen concentration and zooplankton abundance: Deep learning versus traditional regression approach, Ecological Indicators, 100: 99-117. https://doi.org/10.1016/j.ecolind.2018.09.051

Bashiri M,FarshbafGeranmayeh A. 2011. Tuning the parameters of an artificial neural network using central composite design and genetic algorithm, Scientia Iranica, 18(6): 1600-1608. https://doi.org/10.1016/j.scient.2011.08.031

Bashiri M,FarshbafGeranmayeh A. 2013. Simultaneous optimization of effective factors on artificial neural network performance using Box-Behnken design and fuzzy programming, International Journal of Industrial Engineering, 24(1)

Bengio Y. 2012. Practical recommendations for gradient-based training of deep architectures, In: Montavon G. Orr G.B. Müller K.R. (Eds.), Neural Networks: Tricks of the Trade, Springer, 7700: 437-478. https://doi.org/10.1007/978-3-642-35289-8_26

Bischl B, Binder M, Lang M, et al. 2023.Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges, WIREs Data Mining and Knowledge Discovery, 13(2). https://doi.org/10.1002/widm.1484

Box GE, Tiao GC. 2011. Bayesian inference in statistical analysis, John Wiley and Sons, USA

Bull AD. 2011. Convergence rates of efficient global optimization algorithms, Journal of Machine Learning

Research, 12(10): 2879-2904

Chandra R, Tiwari A. 2022. Distributed Bayesian optimisation framework for deep neuroevolution, Neurocomputing, 470: 51-65. https://doi.org/10.1016/j.neucom.2021.10.045

Contal E, Buffoni D, Robicquet A,et al. 2013. Parallel Gaussian process optimization with upper confidence bound and pure exploration, In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 225-240

Falkner S, Klein A, Hutter F. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In: Proceedings of the 35th International Conference on Machine Learning. 1437-1446

Fitri A, Maulud KNA, Rossi F, et al. 2021. Spatial and temporal distribution of dissolved oxygen and suspended sediment in Kelantan River Basin, https://doi.org/10.2991/aer.k.210204.011

Frazier PI. 2018. A tutorial on Bayesian optimization. arXiv, 1807.02811

Gabler V, Wollherr D. 2022. Bayesian optimization with unknown constraints in graphical skill models for compliant manipulation tasks using an industrial robot, Frontiers in Robotics and AI, 9. https://doi.org/10.3389/frobt.2022.993359

Garabaghi FH,Benzer S,Benzer R. 2023. Modeling dissolved oxygen concentration using machine learning techniques with dimensionality reduction approach, Environmental Monitoring and Assessment, 195(7): 879,.https://doi.org/10.1007/s10661-023-11492-3

Glorot X, Bordes A,Bengio Y. 2011,Deep sparse rectifier neural networks, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 315-323

Goodfellow I, Bengio Y, Courville A. 2016. Deep Learning, MIT Press, USA

Hoy ZX, Woon KS, Chin WC, et al. 2022. Forecasting heterogeneous municipal solid waste generation via Bayesian-optimised neural network with ensemble learning for improved generalisation, Computers and Chemical Engineering, 166: 107946. https://doi.org/10.1016/j.compchemeng.2022.107946

Huan J, Cao W, Qin Y. 2018. Prediction of dissolved oxygen in aquaculture based on EEMD and LSSVM optimized by the Bayesian evidence framework, Computers and Electronics in Agriculture, 150: 257-265. https://doi.org/10.1016/j.compag.2018.04.022

Ilievski I, Akhtar T, Feng J,et al. 2017. Efficient hyperparameter optimization for deep learning algorithms using deterministic RBF surrogates, Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). https://doi.org/10.1609/aaai.v31i1.10647

Ionescu G,Frood R,Scarsbrook A,et al. 2023. Impact of training dataset size on technical performance of a deep learning model for detection and quantification of lymphomatous disease on 18F-FDG PET/CT, Journal of Nuclear Medicine, 64(1)

Joy TT, Rana S, Gupta S, et al. 2016.Hyperparameter tuning for big data using Bayesian optimization. 23rd International Conference on Pattern Recognition (ICPR), 2574-2579. https://doi.org/10.1109/ICPR.2016.7900023

Kandasamy K, Krishnamurthy A, Schneider J,et al. 2018.Parallelised Bayesian optimisation via Thompson sampling, International Conference on Artificial Intelligence and Statistics, 133-142

Kannel PR, Lee S, Lee YS, et al. 2007. Application of water quality indices and dissolved oxygen as indicators for river water classification and urban impact assessment, Environmental Monitoring and Assessment, 132(1-3): 93-110. https://doi.org/10.1007/s10661-006-9505-1

Kingma DP, Ba J. 2014. Adam: A method for stochastic optimization, ArXiv, 412.6980

Kumar ZM, Manjula R. 2012. Regression model approach to predict missing values in the Excel sheet databases, International Journal of Computer Science and Engineering Technology, 3(4): 130-135

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning, Nature, 521(7553): 436-444.

https://doi.org/10.1038/nature14539

Lederer J. 2021. Activation functions in artificial neural networks: A systematic overview, ArXiv, 2101.09957v1

Li G, Zhang A, Zhang Q,et al. 2022. Pearson correlation coefficient-based performance enhancement of broad learning system for stock price prediction, IEEE Transactions on Circuits and Systems II: Express Briefs, 69(5): 2413-2417. https://doi.org/10.1109/TCSII.2022.3160266

Liu X,Mihaylova L, George J,et al. 2023. Gaussian process upper confidence bounds in distributed point target tracking over wireless sensor networks, IEEE Journal of Selected Topics in Signal Processing, 17(1): 295-310. https://doi.org/10.1109/JSTSP.2022.3223521

Lu Q, Polyzos KD, Li B, et al. 2023. Surrogate modeling for Bayesian optimization beyond a single Gaussian process, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9): 11283-11296. https://doi.org/10.1109/TPAMI.2023.3264741

Moghadam SV,Sharafati A,Feizi H, et al. 2021. An efficient strategy for predicting river dissolved oxygen concentration: Application of deep recurrent neural network model, Environmental Monitoring and Assessment, 193(12): 798. https://doi.org/10.1007/s10661-021-09586-x

Parsa M, Mitchell JP, Schuman CD, et al. 2020. Bayesian multi-objective hyperparameter optimization for accurate, fast, and efficient neural network accelerator design. Frontiers in Neuroscience, 14. https://doi.org/10.3389/fnins.2020.00667

Pascanu R, Mikolov T,Bengio Y. 2013. On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 1310-1318

Probst P.Boulesteix AL, Bischl B. 2019. Tunability: Importance of hyperparameters of machine learning algorithms. Journal of Machine Learning Research, 20(53): 1-32

Rasmussen CE, Williams, C. 2005. Gaussian Processes For Machine Learning. MIT Press, USA

Rijsdijk J, Wu L, Perin G, et al. 2021. Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(3): 677-707. https://doi.org/10.46586/tches.v2021.i3.677-707

Romero O, Das S, Chen PY, et al. 2020. A dynamical systems approach for convergence of the Bayesian EM algorithm. arXiv, 2006.12690v2

Rouf MA, Islam MJ, Roknuzzaman M, et al. 2022. Vertical profile of dissolved oxygen and associated water variables in the Pasur-Rupsha estuary of Bangladesh. Heliyon, 8(10): e10935. https://doi.org/10.1016/j.heliyon.2022.e10935

Sami Z, Latif SD, Ahmed AN, et al. 2022. Machine learning algorithm as a sustainable tool for dissolved oxygen prediction: A case study of Feitsui Reservoir, Taiwan. Scientific Reports, 12(1): 3649. https://doi.org/10.1038/s41598-022-06969-z

Sherstinsky A. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

Snoek J, Larochelle H,Adams RP. 2012. Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems, 25

Srinivas N, Krause A, Kakade SM, et al. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv, 0912.3995

Steurer M, Hill RJ, Pfeifer N. 2021. Metrics for evaluating the performance of machine learning-based automated valuation models. Journal of Property Research, 38(2): 99-129. https://doi.org/10.1080/09599916.2020.1858937

Sun S. 2013. A review of deterministic approximate inference techniques for Bayesian machine learning. Neural Computing and Applications, 23(7-8): 2039-2050. https://doi.org/10.1007/s00521-013-1445-4

Sutskever I, Vinyals O, Le QV. 2014. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27

Tiyasha T, Tung TM, Bhagat SK, et al. 2021. Functionalization of remote sensing and on-site data for simulating surface water dissolved oxygen: Development of hybrid tree-based artificial intelligence models. Marine Pollution Bulletin, 170: 112639. https://doi.org/10.1016/j.marpolbul.2021.112639

Tran TK, Lee T, Shin JY, et al. 2020. Deep learning-based maximum temperature forecasting assisted with meta-learning for hyperparameter optimization. Atmosphere, 11(5), 487. https://doi.org/10.3390/atmos11050487

Turkey Ministry of Environment and Urban Planning (EUP). 2016. Büyük Menderes NehirHavzasıTaslakYönetimPlanı SÇD Pilot ProjesiStratejikÇevreselDeğerlendirmeRaporu.

Weerts HJP, Mueller AC,Vanschoren J. 2020. Importance of tuning hyperparameters of machine learning algorithms. arXiv, 2007.07588

Wu J, Chen XY, Zhang H, et al. 2019. Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1): 26-40

Xiao X, Yan M, Basodi S, et al. 2020. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. arXiv, 2006.12703

Yan T, Zhou A, Shen SL. 2023. Prediction of long-term water quality using machine learning enhanced by Bayesian optimisation. Environmental Pollution, 318: 120870. https://doi.org/10.1016/j.envpol.2022.120870

Yang J. 2023. Predicting water quality through daily concentration of dissolved oxygen using improved artificial intelligence. Scientific Reports, 13(1): 20370. https://doi.org/10.1038/s41598-023-47060-5

Zhang WJ. 2010. Computational Ecology: Artificial Neural Networks and Their Applications. World Scientific, Singapore

Zhang WJ. 2016. Selforganizology: The Science of Self-Organization. World Scientific, Singapore

Zhang WJ, Liu GH, Dai HQ. 2008. Simulation of food intake dynamics of holometabolous insect using functional link artificial neural network. Stochastic Environmental Research and Risk Assessment, 22(1): 123-133

Zhi W, Feng D, Tsai WP, et al. 2021. From hydrometeorology to river water quality: Can a deep learning model predict dissolved oxygen at the continental scale? Environmental Science and Technology, 55(4), 2357–2368. https://doi.org/10.1021/acs.est.0c06783

Zhu N, Ji X, Tan J, et al. 2021. Prediction of dissolved oxygen concentration in aquatic systems based on transfer learning. Computers and Electronics in Agriculture, 180: 105888. https://doi.org/10.1016/j.compag.2020.105888