Comparative Analysis of Multi-Horizon Deep Learning for Water Quality Forecasting in Shrimp Ponds

Santi[1], Arna Fariza[1,*], Agus Indra Gunawan[2]

[1]Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

[2]Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

Received 18 June 2026; received in revised form 01 August 2026; accepted 03 August 2026

DOI: https://doi.org/10.46604/peti.2026.16540

Abstract

Accurate multi-step water-quality prediction is essential for proactive and sustainable aquaculture management. Existing studies often evaluate only limited deep learning (DL) architectures or forecasting horizons, making fair comparisons difficult. This study systematically compares seven DL architectures for forecasting pH, temperature, salinity, and dissolved oxygen (DO) using 6,741 records collected from shrimp ponds at seven Indonesian locations (2017–2025). All models are evaluated using the same preprocessing, optimization, and evaluation settings for forecasting horizons of 5, 10, 20 and 30 steps ahead. Prediction performance consistently decreases as the forecasting horizon increases. Bidirectional long short-term memory (BiLSTM) achieves the lowest root mean square error (RMSE) (2.18) and mean absolute error (MAE) (0.82) at horizon 5 (H5), although repeated experiments indicate that the observed differences are not statistically significant. Performance varies across longer forecasting horizons, with gated recurrent unit (GRU) achieving the lowest RMSE at H20 (2.69).

Keywords: water quality, forecasting, deep learning, multi-horizon

1. Introduction

Managing water quality is vital in intensive shrimp farming because environmental factors directly affect shrimp growth rates, feed consumption, susceptibility to illness, and overall yield. Vital water quality parameters, such as pH, temperature, salinity, and dissolved oxygen (DO), fluctuate constantly due to environmental changes, feeding patterns, seasonal shifts, water circulation, and pond management [1]. Maintaining these parameters within acceptable levels is essential, as sudden changes can place additional strain on shrimp, reduce survival rates, and negatively affect production outcomes. Constant observation and proactive management are becoming increasingly important as shrimp farming techniques advance to ensure consistent output and reduce operational risks [2].

Monitoring water quality using traditional techniques mostly relies on manual inspections or alarm systems that trigger action only when thresholds are met, often resulting in corrective action only after issues have already surfaced. Even if contemporary sensor-based monitoring systems improve data accessibility by continuous readings, just responding to situations does not sufficiently enable timely interventions and decision-making in fast-changing aquaculture environments [3]. Therefore, using prediction methods to forecast future environmental conditions is necessary to enable proactive management, improve operational efficiency, and minimize the risk of production losses.

* Corresponding author. E-mail address: arna@pens.ac.id

Recent studies have highlighted the growing dependence of agricultural and aquatic production on technology-assisted environmental management. Karakulov et al. [4] reported that the use of climate-smart technologies can increase agricultural productivity and promote environmental sustainability. Similarly, water quality significantly impacts adjacent aquatic ecosystems, emphasizing the need for ongoing environmental monitoring [5]. These findings underscore the need for robust water-quality prediction tools for aquaculture management.

Latest advances in deep learning (DL) have improved forecasting in environmental monitoring applications. Sequential learning approaches can model temporal correlations in sensor-generated data and capture complex nonlinear interactions among environmental factors. DL is especially appealing for aquaculture applications because water-quality parameters change quickly and are closely linked [6]. Even with recent advancements, several limitations remain in current water-quality forecasting research. Past research has primarily focused on assessing single designs rather than conducting thorough comparisons across multiple DL techniques. Moreover, many studies focus on single-parameter prediction challenges or limited forecasting horizons, making it difficult to evaluate model stability across varied forecasting scenarios. Existing comparative research employs varied preprocessing techniques, datasets, and evaluation methods, limiting direct performance comparison across architectures [7]. Various studies have investigated recurrent neural network (RNN), gated recurrent unit (GRU), and long short-term memory (LSTM) models; hybrid bidirectional long short-term memory (BiLSTM), convolutional neural network (CNN)-LSTM, and GRU-LSTM; and attention-based (LSTM-Transformer) architectures for water-quality prediction. Although these methods show encouraging performance improvements, there is still insufficient information to determine which architectures provide the most reliable forecasting for monitoring shrimp ponds across multiple time horizons.

Water-quality forecasting is needed in shrimp farming not only for short-term intervention but also for medium- and long-term operational planning. Short-horizon forecasts facilitate rapid actions such as aeration control, feeding adjustments, and emergency responses. Longer-horizon forecasts support water management, production planning, and resource allocation. Nonetheless, multi-horizon forecasting remains challenging because prediction uncertainty increases with the forecasting horizon [8]. As a result, models that are good at short-term prediction may experience a degradation performance for longer forecasts. Therefore, it is crucial to test the forecasting models on different horizons to build reliable aquaculture monitoring systems. Physics-based hydrodynamic models have also been used for water-quality management. For example, Kim et al. [9] demonstrated that predictive simulation could support operational scenario analysis. However, such models require detailed hydraulic information and differ from data-driven real-time forecasting based on historical sensor observations.

To address these limitations, this study performs a comprehensive and standardized comparison of seven representative deep learning architectures for multi-horizon water-quality forecasting in shrimp ponds. All models are evaluated under identical preprocessing, sequence construction, optimization, hyperparameter, and evaluation settings across four forecasting horizons (H5, H10, H20, and H30). This standardized experimental setting enables a fair and reproducible comparison while minimizing bias caused by differences in datasets and training configurations reported in previous studies. Furthermore, the evaluation uses heterogeneous multi-location shrimp pond datasets, along with repeated experiments and statistical validation, to assess model robustness across different water-quality parameters and forecasting horizons. The main contributions of this study are summarized as follows:

(1) A standardized comparative evaluation of seven representative deep learning architectures under identical preprocessing, optimization, and evaluation settings, enabling fair and reproducible benchmarking for multi-horizon water-quality forecasting.

(2) A comprehensive evaluation using heterogeneous multivariate datasets collected from seven Indonesian shrimp farming locations, providing a more realistic assessment of model generalization under diverse environmental conditions.

(3) A comprehensive benchmark covering H5, H10, H20, and H30 forecasting horizons together with repeated experiments and statistical validation to evaluate model consistency across prediction horizons.

(4) Practical guidance for selecting forecasting architectures according to operational forecasting horizons while balancing predictive performance and computational efficiency.

Although the primary benchmark comprises seven prespecified architectures, PatchTST was additionally evaluated during revision as a supplementary exploratory model. It was included only for contextual comparison and was excluded from the primary statistical hypothesis tests. Other recent architectures, including Autoformer and Informer, remain outside the scope of this study.

2. Literature Review

Water-quality forecasting is an important topic in aquaculture and environmental monitoring. Parameters such as pH, temperature, salinity, and DO are highly dynamic and exhibit strong temporal dependencies, making time-series forecasting methods increasingly important for environmental monitoring [10]. Therefore, this study presents a comparative framework for multi-horizon water-quality forecasting in shrimp ponds using multivariate datasets collected across multiple Indonesian locations. Seven deep learning architectures (RNN, GRU, LSTM, BiLSTM, CNN-LSTM, GRU-LSTM, and LSTM-Transformer) are evaluated under identical preprocessing, optimization, and evaluation settings across forecasting horizons of H5, H10, H20, and H30 in Table 1.

Table 1 Comparative analysis of previous deep learning studies for water quality forecasting

Study	Model used	Key findings	Limitation and gap addressed in this study
Wongburi and Park, 2023[11]	RNN	RNNs captured sequential patterns in water-quality prediction.	Limited architecture and horizon comparison; recurrent models compared across H5–H30.
Xu et al., 2021[12]	GRU	GRU improved prediction performance under sparse, heterogeneous data.	Focused on complex attention optimization; GRU is evaluated under identical settings with other models.
Gao et al., 2023[13]	LSTM	LSTM showed strong multivariate forecasting performance.	Limited parameter and model comparison; LSTM is evaluated using heterogeneous multi-location shrimp pond.
Jamshidzadeh et al., 2024[14]	BiLSTM	Bidirectional learning improved feature extraction.	Forecast windows and indicators were limited; BiLSTM is evaluated across H5, H10, H20, and H30.
Baek et al., 2020[15]	CNN-LSTM	CNN-LSTM captured temporal water-quality variation.	Focused on specific river environments; hybrid models are compared under identical protocols.
Elshewey et al., 2026[16]	GRU-LSTM	GRU-LSTM showed strong temporal feature learning.	Focused on classification; GRU-LSTM is applied to regression-based multi-horizon forecasting.
Huan et al., 2025 [17]	LSTM-Transformer	LSTM-Transformer improved long-range dependency learning.	Computationally complex and feature-dependent; attention-based forecasting is tested under multi-location pond conditions.
This Study	RNN, GRU, LSTM, BiLSTM, CNN-LSTM, GRU-LSTM, LSTM-Transformer	BiLSTM showed competitive short-horizon performance.	Provides a standardized comparison of seven architectures across H5–H30 using heterogeneous Indonesian shrimp pond datasets.

Overall, previous studies demonstrate the effectiveness of deep learning models for water-quality forecasting, with RNN, GRU, LSTM, and hybrid architectures showing different strengths in capturing temporal patterns. However, most studies focus on a single architecture, specific environments, or limited forecasting horizons, making direct performance comparisons difficult. In contrast, this study provides a standardized comparison of seven deep learning architectures across four forecasting horizons (H5, H10, H20, and H30) using heterogeneous shrimp pond datasets from multiple Indonesian locations.

3. Methodology

This section describes the proposed forecasting methodology, including data preprocessing, sequence construction, model training, evaluation, and comparative analysis. Fig. 1 illustrates the overall system design of the proposed internet of things (IoT)-enabled forecasting framework [1].

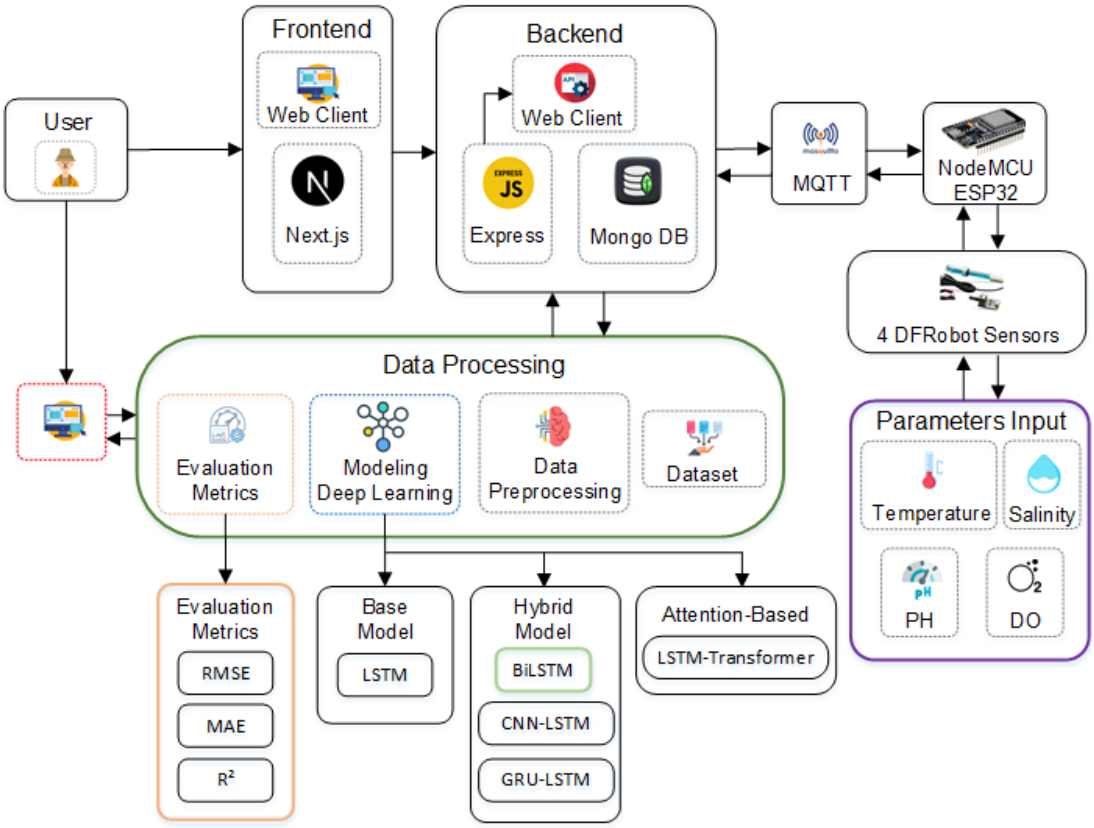


Fig. 1 System design

The workflow starts with real-time data acquisition. For this purpose, an IoT hardware setup with four specific DFRobot physical sensors is used to continuously monitor critical water-quality parameters, i.e., temperature, salinity, pH, and DO. The analog signals generated by these sensors are captured and digitized by a NodeMCU ESP32 microcontroller, which then transmits the processed data packets to the backend server via a lightweight MQTT protocol using a Mosquitto broker [18]. The web application architecture organizes data handling and user interaction upon the arrival of the data stream. The backend engine is built using Express.js and allows data to be ingested through secure application programming interfaces (APIs) and stored in a non-relational MongoDB database to build a comprehensive historical dataset.

The data-processing module preprocesses the data and trains the forecasting models. The processed dataset is then used to evaluate seven architectures across three model groups: baseline recurrent models (RNN, LSTM, and GRU), hybrid models (BiLSTM, CNN-LSTM, and GRU-LSTM), and an attention-based LSTM-Transformer model. The models are evaluated using root mean square error (RMSE) and mean absolute error (MAE) to identify the best architecture for multi-horizon water-quality forecasting. Forecasts are displayed through a Next.js dashboard for decision support. These outputs help farmers make proactive decisions to maintain pond water quality before critical environmental changes.

3.1. Data collections

The dataset consists of multivariate water-quality observations collected from temperature, pH, salinity, and DO sensors. The resulting temporal alignment and schema standardization yields 6,741 observations from 7 Indonesian locations (Bali, Bulukumba, Gresik, Lampung, Probolinggo, Sidoarjo, and Surabaya) between February 2017 and December 2025. The

available observations consisted of 6,355 pH values, 5,816 temperature values, 5,113 salinity values, and 6,532 DO values. Corresponding counts of missing values were 386, 925, 1,628 and 209. Sampling intervals varied among monitoring locations. Bali and Bulukumba predominantly contained daily observations; Lampung had a median interval of approximately six hours, whereas Gresik, Probolinggo, Sidoarjo, and Surabaya exhibited more irregular sampling intervals. Table 2 presents representative samples of the raw water-quality observations collected from all monitoring locations.

Table 2 Raw dataset sample

Location	Pool	Date	Time	Salinity (parts per thousand)	DO (mg/L)	PH	Temp (℃)
Lampung	A1	28/03/2017	-	-	5.20	7.60	27.80
Lampung	B1	01/01/2018	-	-	4.80	7.90	28.00
Probolinggo	P1	25/05/2019	16:10	23.00	7.88	7.82	28.53
Bulukumba	B1	-	-	36.20	5.24	8.75	-
Bali	P1	14/06/2021	-	-	5.70	7.90	-
Gresik	P20.1	07/11/2022	22:53	21.00	4.00	7.00	28.00
Sidoarjo	P28.1	09/06/2023	07.00	25.00	4.57	7.51	24.33
Surabaya	P05	09/05/2024	15:15	-	5.00	6.64	25.25
Sidoarjo	P01	29/04/2025	18:50	32.09	7.03	6.98	23.20

3.2. Data preprocessing pipeline

The suggested approach, as shown in Fig. 2, methodically reduces 15 granular operational steps into 5 consecutive main phases: (i) input harmonization, (ii) temporal validation, (iii) domain-specific cleaning/imputation, (iv) statistical outlier reduction, and (v) sequence creation. No complete records are removed during date validation, duplicate checking, or physical-range validation. Missingness is handled at the variable and sequence levels. Sequences that still contained unavailable historical inputs or target values after preprocessing are excluded during sequence construction. Therefore, preprocessing reduced the number of usable model sequences rather than excluding entire monitoring locations.

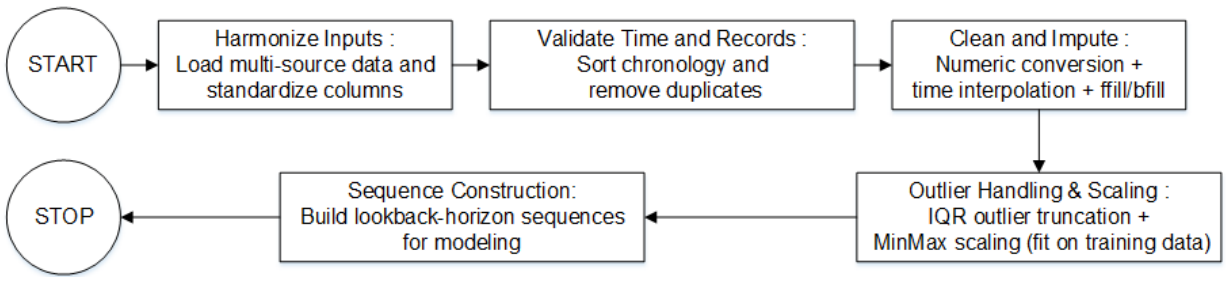


Fig. 2 Preprocessing data

The preprocessing audit confirms that the original dataset contains 6,741 timestamped observations and identifies no invalid timestamps, duplicate location–timestamp records, or physical-range violations. Missing-gap analysis detects 31 gaps spanning more than five consecutive observations, involving 2,788 unresolved parameter values across 2,049 observation rows. All seven monitoring locations contain 3,047–3,122 training sequences, 651–726 validation sequences, and 652–727 testing sequences, depending on the forecasting horizon.

Missing values are handled causally within each location. Short gaps of no more than five consecutive observations are filled using the most recent available historical value, without accessing future observations. Gaps longer than five observations remained unresolved, and any sliding-window sequence containing unresolved input or target values is excluded. The five-observation threshold is selected as a conservative short-gap limit and is additionally examined through supplementary sensitivity experiments.

To safeguard downstream deep learning architectures against instrument noise and extreme variance, an interquartile range (IQR) filter is applied independently to each water quality parameter per city. The numerical IQR thresholds are estimated separately for each water-quality parameter and monitoring location using training observations only. The resulting thresholds are then applied unchanged to the corresponding validation and testing subsets.

Extreme outliers lying outside the statistical boundary are truncated using the following criteria:

$$\text{Lower Bound} = Q_1 - 1.5 \times \text{IQR} \tag{1}$$

$$\text{Upper Bound} = Q_3 + 1.5 \times \text{IQR} \tag{2}$$

where *Q1* and *Q3* represent the 25th and 75th percentiles of the data distribution, respectively, and IQR denotes the interquartile range (IQR = *Q3* - *Q1*). Once statistical stability is achieved, minmax normalization is applied to scale all feature values to a uniform range, thereby mitigating variance caused by disparate physical units of measurement. The transformation is defined as:

$$X_{scaled} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3}$$

where x_{scaled} represents the normalized feature value, x is the original observation, and x_{min} and x_{max} denote the minimum and maximum parameter values, respectively. Crucially, to prevent data leakage and preserve strict experimental validity, the scaling parameters (x_{min} , x_{max}) are fitted solely using the training partition and subsequently utilized to transform both the validation and testing subsets.

The 24-observation lookback is selected to provide sufficient recent temporal context while preserving an adequate number of valid sequences and maintaining manageable computational cost. Because the sampling interval varies across locations, the lookback represents the previous 24 available observations rather than a universal 24-hour period. Supplementary experiments comparing lookbacks of 12, 24, and 48 observations support using 24 observations for the primary benchmark. Overall, the supplementary sensitivity analyses indicate that the principal comparative conclusions remain unchanged across the evaluated lookback configurations, supporting the robustness of the proposed benchmarking framework.

3.3. Workflow of machine learning models

Fig. 3 presents the modeling workflow. The chronologically ordered data are divided into 70% training, 15% validation, and 15% testing subsets. Sliding-window sequences are constructed using a 24-observation lookback for H5, H10, H20, and H30 forecasting. Random shuffling is turned off to preserve temporal order.

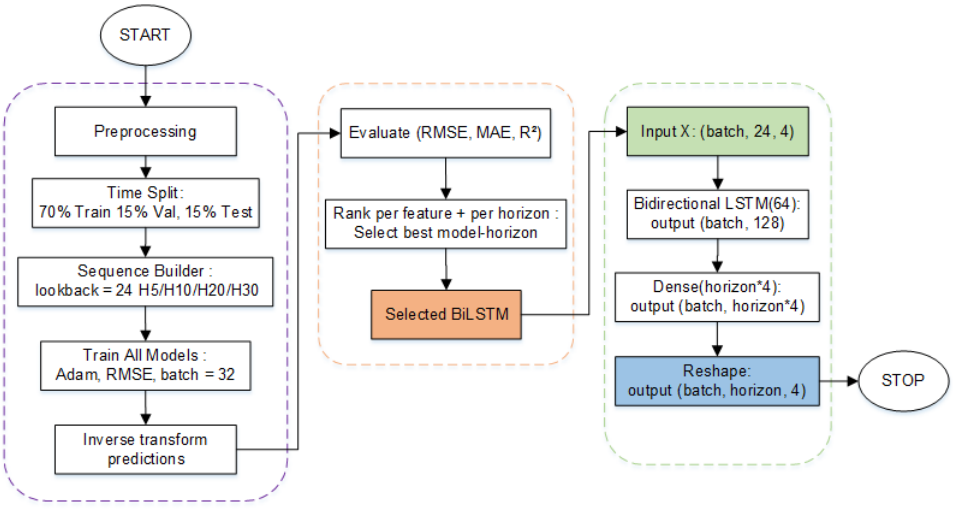


Fig. 3 The workflow of models' prediction

All architectures are trained using the Adam optimizer, MSE loss, and a maximum of 30 epochs. Early stopping with a patience of 10 epochs is based on validation loss, and the best validation weights were restored. Candidate learning rates of 3×10^{-4} and 10^{-3}, and batch sizes of 24, 32, and 64, are evaluated on the same validation subset for each model. The selected common configuration used a batch size of 32. Predictions are inverse-transformed to their original physical units and evaluated using RMSE, MAE, and coefficient of determination (R^2). BiLSTM is selected for detailed visualization because it produces the lowest empirical errors at the short forecasting horizons.

3.4. Performance metric indicators

To evaluate the performance of the proposed prediction model, three evaluation metrics are employed: RMSE, MAE, and R^2.

(1) MAE calculates the average absolute difference between predicted values and actual values. Unlike RMSE, MAE treats all errors equally without emphasizing large errors [19]. The MAE is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{4}$$

where y_i is the actual observed value, $\hat{y}$ represents the predicted value, $\bar{y}$ is the mean of actual values, and n is the total number of samples.

(2) RMSE measures the square root of the average squared differences between actual values and predicted values [20]. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{5}$$

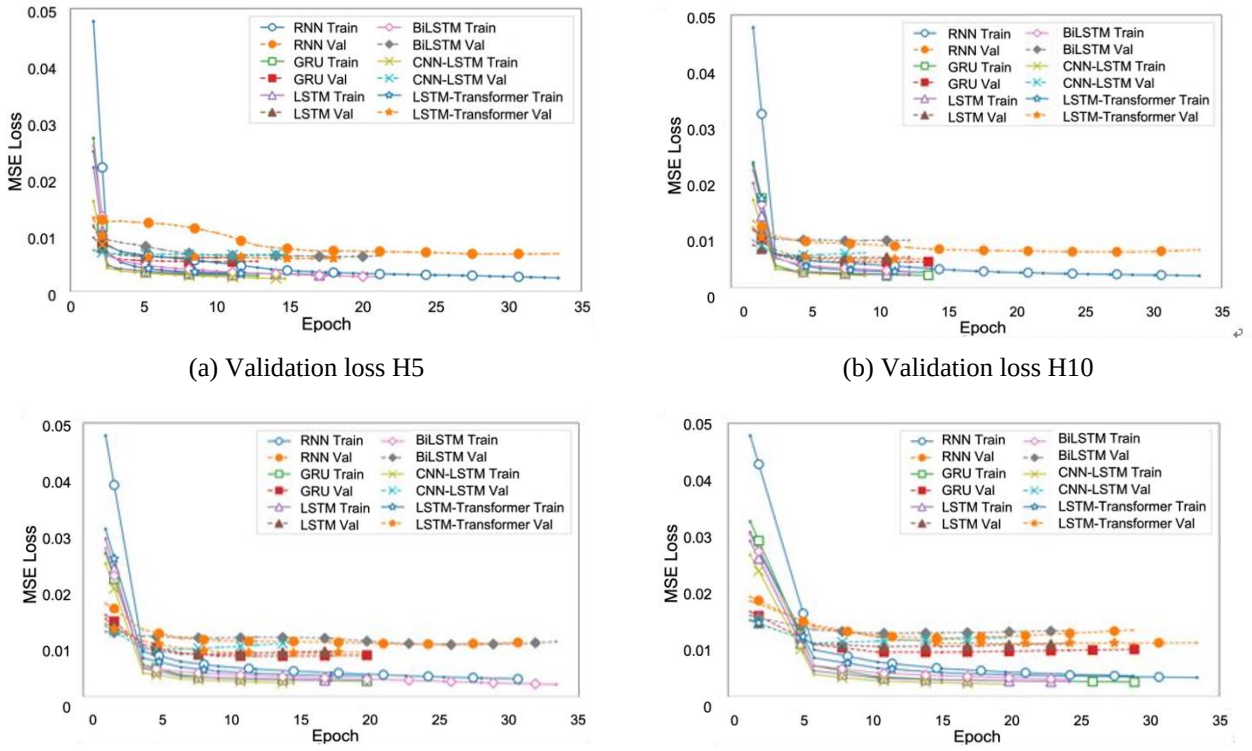
(3) R^2 measures how well the prediction model explains the variance of the actual data. The value of R^2 ranges from 0 to 1, where values closer to 1 indicate better model performance [21]. The coefficient of determination is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

4. Result and Discussion

This section presents the forecasting results and compares model performance across H5, H10, H20, and H30 horizons. Fig. 4 illustrates the validation loss, measured using MSE, for each forecasting horizon. Overall, validation loss increased with longer forecasting horizons, as indicated by the upward shift in MSE values from approximately 0.0150 at H5 to over 0.0200 at H30. Among the evaluated architectures, recurrent neural networks (RNN, LSTM, and GRU) and their hybrid variants (BiLSTM, CNN-LSTM, and GRU-LSTM) demonstrate rapid convergence, typically stabilizing within the first 5 to 10 epochs across all horizons.

Notably, the GRU shows competitive validation loss across several longer horizons, indicating stable generalization capability. Conversely, the LSTM-Transformer hybrid exhibits a distinctly slower convergence rate, requiring significantly more epochs (up to 28 epochs in H5) to reach stability. While the LSTM-Transformer underperforms at shorter horizons (H5 and H10), its performance gap narrows at longer horizons (H20 and H30). However, it generally maintains a higher validation loss than the more compact recurrent and hybrid models. Early stopping mechanisms are also apparent: training for simpler architectures automatically terminates early when validation loss stagnates, preventing overfitting.



(a) Validation loss H5

(b) Validation loss H10

(c) Validation loss H20

(d) Validation loss H30

Fig. 4 MSE validation loss across horizons

4.1. Performance metrics comparison

The models' performance is evaluated using the metrics RMSE, MAE, and R^2 for all the forecast horizons to compare in detail the predictive accuracy of the seven deep learning frameworks. The evaluation comprises four forecast horizons (H5, H10, H20, and H30) to enable evaluation of the models in both short-term and long-term prediction settings with identical preprocessing, training, and evaluation parameters. The summary of model performance across forecasting horizons forms the basis for the comparative analysis. Table 3 presents the comparative forecasting performance of all deep learning architectures based on RMSE, MAE, and R2 across four prediction horizons: H5, H10, H20, and H30.

Table 3 Performance metrics evaluation

Model	Horizon	RMSE	MAE	R2
RNN	H5	2.67	1.30	0.51
	H10	2.99	1.49	0.41
	H20	2.80	1.37	0.30
	H30	2.96	1.48	0.26
GRU	H5	2.41	0.95	0.53
	H10	2.67	1.37	0.42
	H20	2.69	1.45	0.31
	H30	2.95	1.68	0.20
LSTM	H5	2.32	0.92	0.56
	H10	2.64	1.24	0.36
	H20	2.84	1.57	0.27
	H30	2.88	1.52	0.28
BiLSTM	H5	2.18	0.82	0.54
	H10	2.59	1.21	0.44
	H20	2.77	1.40	0.32
	H30	2.91	1.51	0.25

Table 3 Performance metrics evaluation (continued)

Model	Horizon	RMSE	MAE	R2
CNN-LSTM	H5	2.45	1.08	0.46
	H10	2.63	1.22	0.36
	H20	2.85	1.48	0.32
	H30	2.91	1.55	0.23
GRU-LSTM	H5	2.48	1.10	0.55
	H10	2.72	1.33	0.40
	H20	2.82	1.49	0.26
	H30	2.94	1.55	0.25
LSTM-Transformer	H5	2.50	1.20	0.43
	H10	2.78	1.43	0.20
	H20	3.08	1.76	0.23
	H30	3.40	1.91	0.11

In summary, the performance of most models deteriorates with increasing forecasting horizon, as indicated by increases in RMSE and MAE and decreases in R2 scores, suggesting that long-term multivariate water quality forecasting is more difficult than short-term prediction. BiLSTM obtains with values of RMSE = 2.18, MAE = 0.82, and $R^2 = 0.54$ at H5. For longer horizons, the models' performance is more mixed. GRU records the lowest RMSE of 2.69 at H20, while BiLSTM remained competitive with RMSE = 2.77 and MAE = 1.40. At H30, LSTM achieves the lowest RMSE (2.88) and the highest R2, while BiLSTM remains competitive with a comparable RMSE (2.91) and a better MAE (1.51).

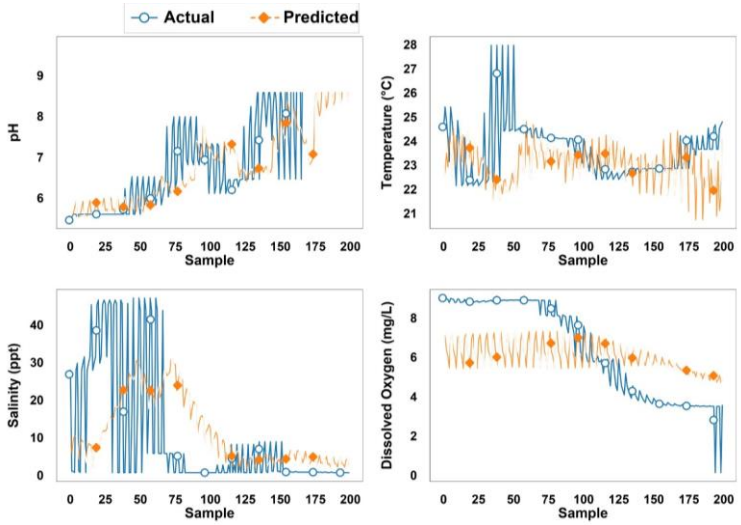
The conventional RNN and LSTM models performed satisfactorily, but the best-performing architectures generally outperformed them. In several cases, hybrid models such as CNN-LSTM and GRU-LSTM achieve comparable results, but they are not consistently better than other models across all forecast horizons. The worst performance was observed in the LSTM-Transformer across all experiments, with an increase in RMSE from 2.50 at H5 to 3.40 at H30 and a decrease in R2 from 0.43 to 0.11, indicating its limited ability to perform long-horizon forecasting under the configuration used in this study. These findings indicate that short-term forecasting is generally more reliable, although BiLSTM do not show statistically significant superiority.

The observed performance differences suggest that forecasting accuracy depends not only on model architecture but also on the interaction between model complexity and dataset characteristics. More complex architectures, such as the LSTM-Transformer model, do not consistently improve prediction accuracy under the present experimental conditions, possibly because the available training sequences and the 24-observation input window provide limited opportunity to exploit their greater modeling capacity. In contrast, the recurrent and hybrid architectures achieve a more favorable balance between predictive performance and model complexity across the evaluated forecasting horizons.

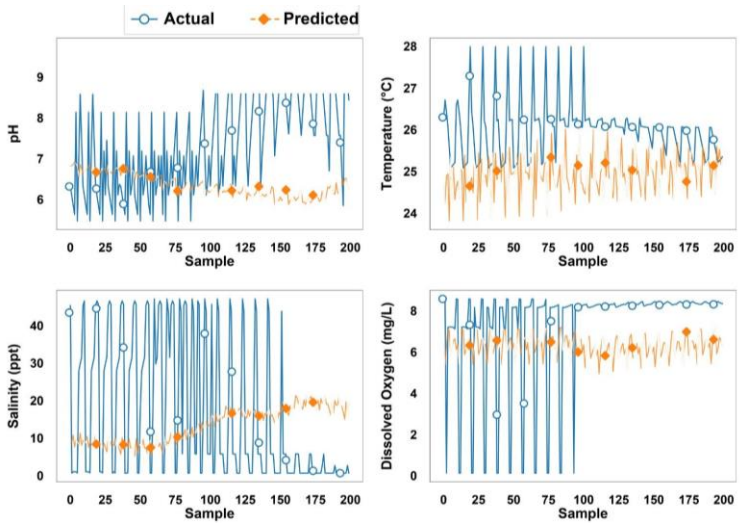
4.2. Prediction visualization

Fig. 5 presents a qualitative comparison between the actual values and BiLSTM-based predictions across four forecasting horizons (H5, H10, H20, and H30) for pH, temperature, salinity, and DO. The visual results support the quantitative evaluation, showing that the model performs better at shorter horizons and gradually loses predictive precision as the forecasting horizon increases. At H5, the predicted curves generally follow the observed patterns, particularly for temperature and DO, indicating that BiLSTM can capture the dominant temporal structure of relatively smoother variables. At H10, the predictions still approximate the general trend, although deviations become more visible during abrupt changes and high-frequency fluctuations. The main limitation shown in the figure is the model's tendency to smooth, which reduces its responsiveness to sharp peaks, sudden drops, and extreme variations. This behavior is especially evident for pH and salinity; whose actual series exhibit stronger variability across samples. At H20 and H30, the predicted curves become flatter and more centered around

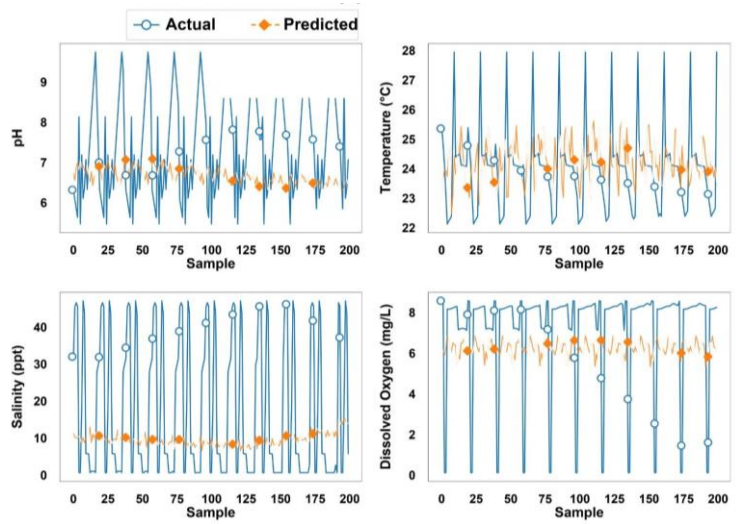
the historical mean, indicating a reduced ability to preserve feature-level variability in longer-horizon forecasting. Overall, the qualitative evaluation confirms that BiLSTM is more reliable for short-term water-quality forecasting, while its performance decreases for highly dynamic variables and extended prediction horizons.



(a) Water quality prediction for H5

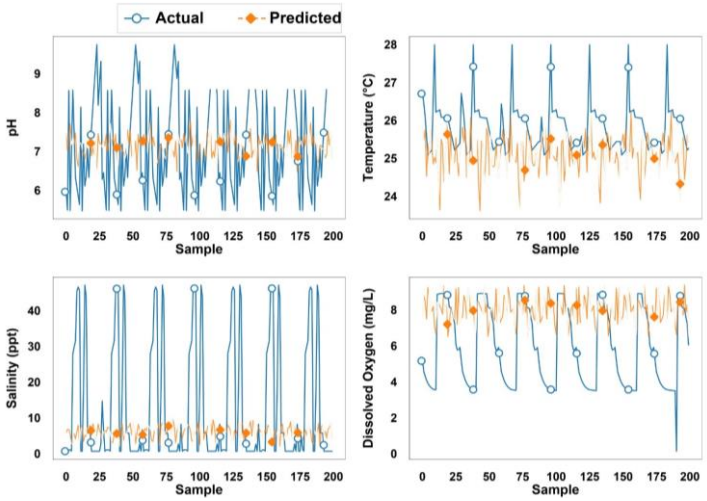


(b) Water quality prediction for H10



(c) Water quality prediction for H20

Fig. 5 Water quality prediction results the BiLSTM model



(d) Water quality prediction for H30

Fig. 5 Water quality prediction results the BiLSTM model (continued)

4.3. Repeated experiment and statistical validation

To assess the robustness of the primary benchmark, the seven prespecified models are evaluated at four forecasting horizons using five independent random seeds (42–46). PatchTST is additionally evaluated during revision as a supplementary exploratory model. It is displayed in Figs. 6 and 7 for contextual comparison but is excluded from the primary Friedman and pairwise Wilcoxon statistical analyses. All runs used identical chronological partitions, preprocessing procedures, hyperparameter settings, and evaluation protocols.

Fig. 6 summarizes the mean performance obtained from five independent runs across the sixteen parameter–horizon forecasting tasks. The seven primary benchmark models are shown together with PatchTST, which is added during revision as an exploratory reference model. PatchTST is included in the figure to illustrate its relative forecasting behavior under the same dataset and preprocessing conditions; however, it is not included in the primary statistical hypothesis tests. The statistical comparison reported in this study therefore refers only to the seven prespecified benchmark architectures.

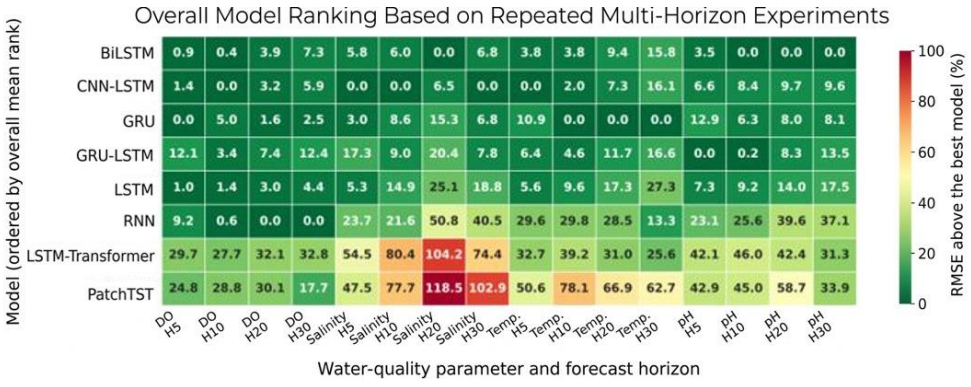


Fig. 6 Repeated-run performance comparison

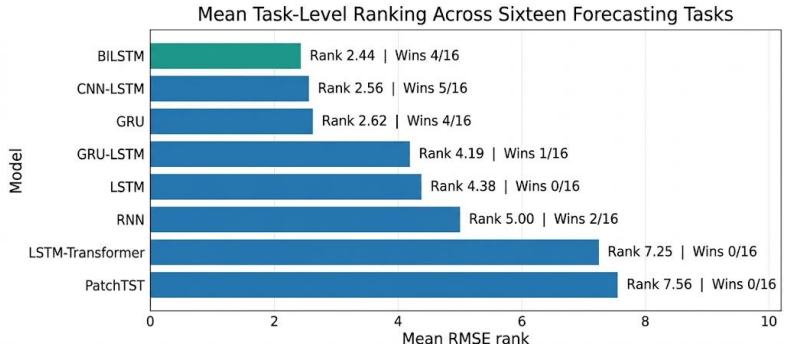


Fig. 7 Mean task-level ranking across sixteen forecasting tasks

Fig. 7 presents the mean RMSE rank across the sixteen forecasting tasks. The ranking values shown for PatchTST are descriptive and are provided only for exploratory comparison. The conclusions concerning model consistency, statistical significance, and the absence of statistically confirmed superiority are derived from the seven primary benchmark models.

Across the five repeated runs, all benchmark models showed relatively stable performance. The Friedman test indicated overall differences among the seven benchmarked architectures. However, none of the pairwise Wilcoxon signed-rank tests remained statistically significant after Holm correction. These results suggest that the empirical ranking should be interpreted as comparative rather than statistically conclusive.

Table 4 Computational complexity comparison

Model	Median parameters	Mean training time
RNN	8,316	5.13 s
GRU	17,340	10.34 s
LSTM	21,564	11.30 s
CNN-LSTM	37,756	9.11 s
BiLSTM	43,068	18.54 s
GRU-LSTM	50,360	23.91 s
LSTM-Transformer	55,036	26.90 s
PatchTST	436,284	57.56 s

Table 4 summarizes the computational complexity of the benchmark models based on median parameter count and mean training time. Although PatchTST contained substantially more parameters and required considerably longer training time, it did not consistently outperform the recurrent and hybrid architectures, indicating that increased model complexity alone does not guarantee superior forecasting performance for the present dataset.

5. Conclusions

This study systematically evaluated seven prespecified recurrent, hybrid, and attention-based architectures for multi-horizon multivariate water quality forecasting. PatchTST was also evaluated as a supplementary exploratory model but was excluded from the primary statistical comparison. The main findings are summarized as follows:

(1) Forecasting performance generally declined with increasing prediction horizons, confirming that long-range multivariate water quality forecasting is more challenging than short-term prediction.

(2) H5 and H10 yielded more reliable forecasts, indicating their suitability for operational shrimp pond monitoring and timely decision-making.

(3) BiLSTM achieved the lowest empirical errors at H5 and H10, although its advantage was not statistically significant after Holm correction.

(4) Recurrent and hybrid architectures generally outperformed the LSTM-Transformer configuration, suggesting that greater architectural complexity does not necessarily improve forecasting performance.

(5) DO and temperature were more predictable than salinity and pH, with the latter showing greater forecasting difficulty due to higher variability and nonlinear behavior.

From a practical perspective, recurrent and hybrid architectures offer a favorable balance between predictive performance and computational efficiency, making them suitable candidates for resource-constrained shrimp pond monitoring systems. Future work will extend the benchmark to recent transformer-based architectures, including Autoformer, Informer, and TimesNet, using the same standardized experimental protocol.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] K. Zhang, Z. Ye, M. Qi, W. Cai, J. L. Saraiva, Y. Wen, et al., "Water Quality Impact on Fish Behavior: A Review from an Aquaculture Perspective," Reviews in Aquaculture, vol. 17, no. 1, article no. e12985, 2025.

[2] M. G. C. Emerenciano, A. N. Rombenso, F. D. N. Vieira, M. A. Martins, G. J. Coman, H. H. Truong, et al., "Intensification of Penaeid Shrimp Culture: An Applied Review of Advances in Production Systems, Nutrition and Breeding," Animals, vol. 12, no. 3, article no. 236, 2022.

[3] L. Liu, W. Cheng, and H. W. Kuo, "A Narrative Review on Smart Sensors and IoT Solutions for Sustainable Agriculture and Aquaculture Practices," Sustainability, vol. 17, no. 12, article no. 5256, 2025.

[4] F. Karakulov, B. Menglikulov, A. Mamasadikov, S. Kholbutaeva, A. Mukhtorov, G. Abdulkhaeva, et al., "The Impact of Climate-Smart Technology Adoption on Agricultural Productivity and Environmental Sustainability," Emerging Science Journal, vol. 9, no. 6, pp. 3087-3115, 2025.

[5] L. Salazar-Gámez, G. Cárdenas-Calvachi, M. Bolaños-Bolaños, M. Paz-Solarte, A. Bernal-Meneses, J. Camues-Sanchez, et al., "Impact of Water Quality and Sediments on the Riparian Vegetation of Andean Lake," Civil Engineering Journal, vol. 11, no. 9, 2025.

[6] T. Li, J. Lu, J. Wu, Z. Zhang, and L. Chen, "Predicting Aquaculture Water Quality Using Machine Learning Approaches," Water, vol. 14, no. 18, article no. 2836, 2022.

[7] M. H. Nishat, M. H. R. B. Khan, T. Ahmed, S. N. Hossain, A. Ahsan, M. M. El-Sergany, et al., "Comparative Analysis of Machine Learning Models for Predicting Water Quality Index in Dhaka's Rivers of Bangladesh," Environmental Sciences Europe, vol. 37, no. 1, article no. 31, 2025.

[8] Y. Lin, "Progressive Neural Network for Multi-Horizon Time Series Forecasting," Information Sciences, vol. 661, article no. 120112, 2024.

[9] S. Kim, "Sluice Gate Operation and Managed Water Levels Improve Predicted Estuarine Lake Water Quality," Civil Engineering Journal, vol. 11, no. 1, pp. 178-189, 2025.

[10] X. Ji, L. Liu, B. Duan, Y. Li, H. Xing, B. Wang, et al., "Long-Term Multivariate Water Quality Forecasting for Sustainable Aquaculture Management," Water Research X, vol. 29, article no. 100402, 2025.

[11] P. Wongburi and J. K. Park, "Prediction of Wastewater Treatment Plant Effluent Water Quality Using Recurrent Neural Network (RNN) Models," Water, vol. 15, no. 19, article no. 3325, 2023.

[12] J. Xu, K. Wang, C. Lin, L. Xiao, X. Huang, and Y. Zhang, "FM-GRU: A Time Series Prediction Method for Water Quality Based on Seq2seq Framework," Water, vol. 13, no. 8, article no. 1031, 2021.

[13] Z. Gao, J. Chen, G. Wang, S. Ren, L. Fang, A. Yinglan, et al., "A Novel Multivariate Time Series Prediction of Crucial Water Quality Parameters with Long Short-Term Memory (LSTM) Networks," Journal of Contaminant Hydrology, vol. 259, article no. 104262, 2023.

[14] Z. Jamshidzadeh, M. Ehteram, and H. Shabanian, "Bidirectional Long Short-Term Memory (BILSTM) - Support Vector Machine: A New Machine Learning Model for Predicting Water Quality Parameters," Ain Shams Engineering Journal, vol. 15, no. 3, article no. 102510, 2024.

[15] S. S. Baek, J. Pyo, and J. A. Chun, "Prediction of Water Level and Water Quality Using a CNN-LSTM Combined Deep Learning Approach," Water, vol. 12, no. 12, article no. 3399, 2020.

[16] A. M. Elshewey, H. M. El-Bakry, and A. M. Osman, "Enhancing Water Potability Classification Using a Hybrid GRU with LSTM Deep Learning Models Based on Relief Algorithm and Bayesian Optimization," Evolving Systems, vol. 17, no. 1, article no. 10, 2026.

[17] J. Huan, C. Zhang, X. Xu, Y. Qian, H. Zhang, Y. Fan, et al., "River Water Quality Forecasting: A Novel LSTM-Transformer Approach Enhanced by Multi-Source Data," Environmental Monitoring and Assessment, vol. 197, no. 9, article no. 1040, 2025.

[18] R. P. Shete, A. M. Bongale, and D. Dharrao, "Lightweight Cryptographic and Scalable IoT Systems for Encryption across GSM-MQTT Architectures in Resource-Constrained Aquaculture Environment," Engineering, Technology & Applied Science Research, vol. 15, no. 4, pp. 25133-25139, 2025.

[19] S. M. Robeson and C. J. Willmott, "Decomposition of the Mean Absolute Error (MAE) Into Systematic and Unsystematic Components," PLOS ONE, vol. 18, no. 2, article no. e0279774, 2023.

[20] D. S. K. Karunasingha, "Root Mean Square Error or Mean Absolute Error? Use Their Ratio as Well," Information Sciences, vol. 585, pp. 609-629, 2022.

[21] D. Zhang, "Coefficients of Determination for Mixed-Effects Models," Journal of Agricultural, Biological, and Environmental Statistics, vol. 27, no. 4, pp. 674-689, 2022.