Journal of Artificial Intelligence and Technology, 2026, 6, 977-988
https://doi.org/10.37965/jait.2026.1475

ISTP
RESEARCH ARTICLE

Short-Term Water Quality Index Forecasting Using Bayesian-Optimized Machine Learning and Hybrid Time-Series Features

Iffati Feruzia Putri, Giva Andriana Mutiara, and Muhammad Rizqy Alfarisi
Department of Applied Science, Telkom University, Bandung, Indonesia

(Received 04 June 2026; Revised 18 September 2026; Accepted 29 September 2026; Published online 30 October 2026)

Abstract: Freshwater aquaculture environments depend on strict water quality standards, but traditional monitoring is point-in-time and existing literature treats water quality largely as static classification, leaving high-frequency temporal forecasting a significant gap. This study investigates high-resolution forecasting of the Weighted Arithmetic Water Quality Index (WAWQI) using hybrid time-series features and Bayesian-optimized machine learning on data recorded at a nominal 15-s interval in an aquaculture lake, across five horizons from 1 to 60 min. Seven algorithms are compared against persistence, AutoRegressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) baselines on identical rows under a leakage-controlled protocol (chronological split, embargo, single test-set reading), and every comparison is tested with the Diebold–Mariano test and corrected for multiple testing. Persistence alone attains a coefficient of determination of 0.950 at 1 min, falling to −0.214 at 60 min, so goodness-of-fit statistics reported without a naive reference are uninformative. Of 112 model–baseline comparisons, one improves significantly on persistence. Rolling-origin evaluation shows forecasting skill is conditional on target variability, reaching a skill score of 0.503 at 60 min in the most variable test window and −0.065 in the least variable (zero indicates no improvement on the last observation). Permutation importance on held-out data ranks dissolved oxygen (DO), not electrical conductivity, first among sensor channels, and variance decomposition attributes 67.7 % of index variance to DO and 24.5 % to turbidity. These results establish the conditions under which high-frequency forecasting adds information beyond persistence.

Keywords: Bayesian Optimization; ensemble machine learning; forecast evaluation; high-frequency data; time-series analysis; water quality forecasting; WAWQI

I. INTRODUCTION

Freshwater ecosystems and aquaculture environments depend on strict water quality standards to prevent aquatic mortality and ecological degradation [1,2]. The Weighted Arithmetic Water Quality Index (WAWQI) aggregates multiple sensor parameters into a single composite score, providing a standardized metric for water health [2,3]. Traditional WAWQI assessment requires manual sampling and laboratory analysis, whereas Internet of Things (IoT) instrumentation enables continuous in situ observation of the same parameters [4,5], generating high-frequency multivariate time series that machine learning has been widely applied to interpret [6,7].

Existing literature nonetheless approaches WAWQI prediction largely as a classification task or a static regression problem [8], on datasets recorded at daily, monthly or seasonal intervals, and minute-level forecasting remains a significant gap [9]. Chaotic dynamics have been reported in hydrological time series [10,11]; characterizing them requires records considerably longer than a single short deployment provides, so this study does not adopt that framing and instead measures the limits of short-horizon predictability directly, through the decay of the index's own autocorrelation. Reactive classification determines the current state of water quality but cannot anticipate degradation, and previous research on this specific aquaculture environment produced an accurate static classification model while identifying temporal forecasting as an essential requirement for deployment [8].

Using an IoT dataset collected at a nominal 15-s interval from an active aquaculture lake, this study compares seven algorithms across five horizons from 1 to 60 min, together with a persistence baseline, ARIMA, LSTM, and GRU evaluated on identical rows. The contributions are (i) the adaptation of the static WAWQI formulation into a time-series forecasting target at sub-minute resolution, defined as the h-step change in the index rather than its level; (ii) a leakage-controlled evaluation protocol in which every model is compared against a persistence baseline on identical rows, every comparison is tested for significance, and the resulting p-values are corrected for multiple testing; and (iii) an identification of the sensor parameters that drive short-term index variation using permutation importance computed on held-out data, with a feature ablation quantifying what the engineered features contribute. The second requires emphasis: on a target this strongly autocorrelated, the last observation is already a strong forecast, so the question is not whether a model attains a high goodness-of-fit score but whether it improves on that baseline by a margin that survives testing.

II. RELATED WORKS

A. MACHINE LEARNING FOR WATER QUALITY INDEX PREDICTION

Ahmed *et al.* [6] showed that supervised learning, particularly Gradient Boosting, outperforms traditional regression in estimating

Corresponding authors: Giva Andriana Mutiara (e-mail: givamz@telkomuniversity.ac.id); Iffati Feruzia Putri (e-mail: iffati.feruzia22@gmail.com); Muhammad Rizqy Alfarisi (e-mail: mrizkyalfarisi@telkomuniversity.ac.id).

© The Author(s) 2026. This is an open access article published under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

water quality indices, and ensemble methods have since become dominant: Choudhary *et al.* [7] reported $R^2 = 0.9952$ for static WAWQI prediction using a stacked ensemble on Indian river data, and Doan and Nguyen [12] reported 99.17–99.59 % accuracy across multiple WQI formulations with SHapley Additive exPlanations (SHAP)-identified conductivity and pH as top contributors. Each of these operates on periodic, non-sequential samples, treating observations as independent; temporal structure is not modeled. A point that recurs across this literature bears directly on the present work: the index being predicted is not a fixed quantity. Chidiac *et al.* [3] review the development of water quality indices and identify weight selection as a persistent source of divergence between formulations, so two indices computed from the same readings under different weightings are different series with different variance structure and different dominant parameters.

B. IoT MONITORING AND TEMPORAL FEATURES

Baena-Navarro *et al.* [4] predicted dissolved oxygen (DO) with $R^2 = 0.999$ from an IoT multiparameter system in a freshwater aquaculture facility, and Lal *et al.* [5] developed a low-cost lake monitoring system at high temporal resolution. Mutiara *et al.* [8] classified water quality degradation in the same lake studied here with an identical seven-parameter configuration and identified short-term forecasting as the necessary next step. On feature construction, Safat *et al.* [13] showed that lag variables with rolling means improve precipitation prediction, and Lorente-González *et al.* [14] found autoregressive lags improved DO models in coastal lagoons; Su *et al.* [15] evaluated hybrid ARIMA–Multilayer Perceptron (MLP) models for river water quality, and Willard *et al.* [9] observed that sub-hourly forecasting remains among the least studied configurations in the literature.

C. OPTIMIZATION AND FORECAST EVALUATION

Yang and Shami [16] confirmed that Bayesian Optimization with a Tree-structured Parzen Estimator is more efficient than grid or random search in high-dimensional spaces, and Optuna [17] implements it. The evaluation protocol used here comes from the forecasting literature rather than the water quality literature, where it is applied less often. Bergmeir and Benítez [18] show that cross-validation for time-series predictors must respect temporal order, and Kapoor and Narayanan [19] catalog across 17 fields how information leaking from the evaluation set into model selection produces over-optimistic results that ordinary error metrics do not reveal. Beck *et al.* [20] show that for series of low intrinsic predictability a model can appear successful unless compared rigorously against a naive forecast, precisely the condition of a high-frequency water quality index. Such comparisons are conventionally tested with the Diebold–Mariano statistic [21] under the small-sample correction of Harvey *et al.* [22], with p-values adjusted for the family-wise error rate [23] or the false discovery rate [24] when many tests are run. Zeng *et al.* [25] report that simple linear baselines match or exceed far more complex sequence architectures across forecasting benchmarks, motivating the inclusion of both a regularized linear model and recurrent baselines.

III. METHODOLOGY

Figure 1 shows the workflow: sensor acquisition and preprocessing, index construction, forecasting formulation and feature engineering, model selection under a leakage-controlled protocol, and evaluation against baselines with significance testing.

A. STUDY AREA AND DATA COLLECTION

Data were collected from a freshwater lake at Telkom University, Bandung, Indonesia (6.970°S, 107.630°E). The lake supports active fish farming and functions as a campus stormwater retention basin, receiving organic loading from feed residue, metabolic waste, and urban runoff; limited circulation promotes stratification and oxygen depletion, and paddlewheel aerators are operated manually. The same site was used by Mutiara *et al.* [8]. A floating IoT node built around an ESP32-S3 microcontroller recorded seven parameters and transmitted them to a database (Fig. 2): temperature, pH, DO, turbidity, electrical conductivity (EC), total dissolved solids (TDS), and oxidation–reduction potential (ORP). The seven probes are shown in Fig. 3 and their manufacturer specifications in Table I; all are DFRobot Gravity-series units, so the measurement ranges are those of low-cost commercial instrumentation rather than laboratory reference equipment, which bears on the reliability discussion in Section IV.E.

1). SAMPLING INTERVAL. The firmware sets a nominal 15-s delay between readings, but the realized interval is longer and alternates between 15 s and 17 s, with a mean of 16.00 s ($n = 10{,}981$, sd = 1.06 s); those two values account for 98.1 % of intervals. The excess is deterministic rather than random: the

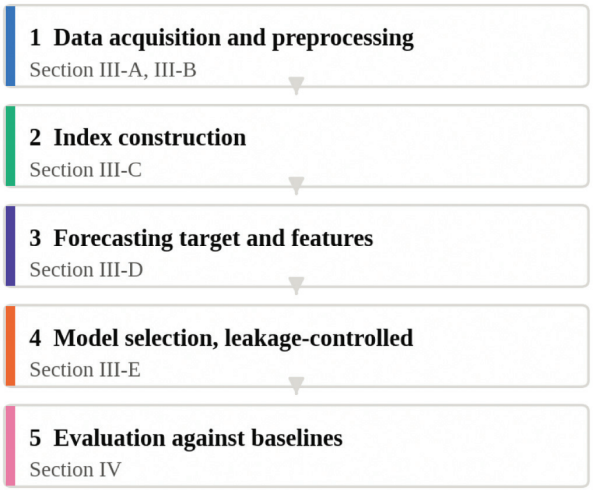


Fig. 1. Workflow of the study. Stages 1 to 4 correspond to Sections III.A to III.E; the evaluation stage corresponds to Section IV.

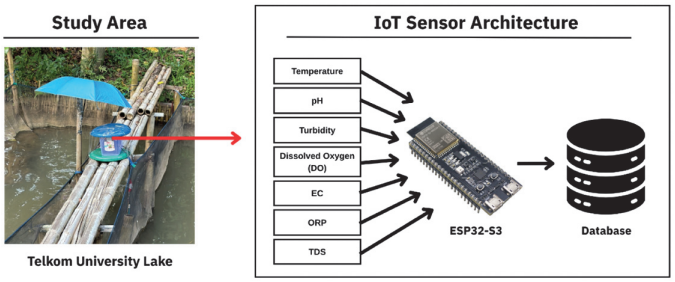


Fig. 2. Study area and IoT sensor architecture. Seven probes are read by an ESP32-S3 node and written to a database.

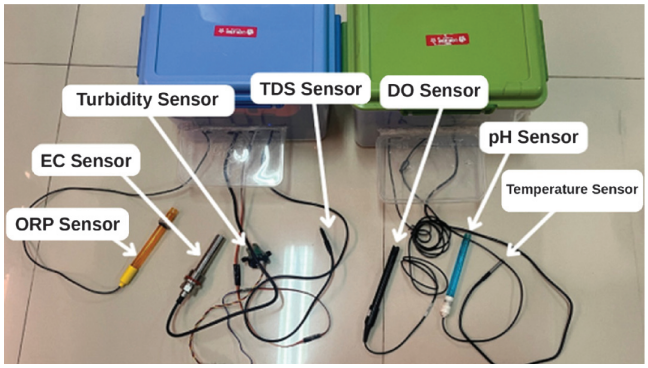


Fig. 3. The seven physicochemical probes deployed at the site.

acquisition loop uses a blocking delay, so the nominal interval is imposed *in addition to* the time taken by seven sequential sensor reads, an LCD refresh and WiFi transmission, instead of being the period between the starts of successive cycles. Horizons are specified at the **nominal 15-s interval**, so $h = 4$, 20, 60, 120, and 240 steps correspond exactly to 1, 5, 15, 30, and 60 min of design time; each horizon therefore spans about 7 % more wall-clock time than its label, which affects the interpretation of the labels but no comparison, since every method is evaluated on identical rows.

2). RECORD. Recording began on 12 May 2026 and produced 27,516 raw readings in two sessions separated by a power interruption, spanning approximately 115 hours at the nominal interval; at the realized interval of 16.00 s, measured over the timestamped portion of the record, it spans approximately 122 hours, and the true wall-clock span is longer than either because recording gaps are counted in neither. Correct wall-clock timestamps are available only to 14 May 2026 at 15:04, at which point the interruption reset the real-time clock to its power-on default rather than stopping it, leaving correct times for 10,984 of the 16,980 analysis rows (65 %) and well-formed but incorrect timestamps for the remainder. Gaps longer than 60 s divide the record into five continuous blocks, three substantial (5,601, 5,365, and 5,920 rows) and two short remnants. Removing 36 stabilization rows, 76 empty records and 13 retrieval rows leaves 27,391. The analysis window is the first 17,100 rows; after removing 120 post-interruption transient rows, it comprises **16,980 observations**, spanning 70.8 hours at the nominal interval and 75.5 hours at the realized one. The remainder is excluded because a sensor cleaning event changes the turbidity regime, from a mean of 13.29 NTU in the analysis subset to 4.38 NTU in the excluded remainder. That makes the two periods different measurement conditions rather than different water conditions. Section IV.D reports what happens when the full record is used. Table II gives descriptive statistics.

B. PREPROCESSING

Four mechanisms are applied, with per-channel counts in Table III. Values outside each sensor's operating range are treated as **missing rather than clipped** because clipping preserves the row count but manufactures a pile-up of observations sitting exactly on the limit which propagates into the index. Short-lived spikes are identified by a 7-point rolling median with a threshold on the median-absolute-deviation z-score, set at **6.0** from the data rather than by convention: raising the threshold until the short-term variability of the index stops growing gives std(ΔWAWQI, 4 steps) = 1.95 at 2.5, **2.16 at 6**, and

Table I. Sensor specifications (DFRobot Gravity series)

Parameter	Product	Measurement range	Interface
Temperature	Waterproof DS18B20 Kit (KIT0021)	−55 to 125°C (±0.5°C, −10 to 85°C)	Digital 1-Wire; 3.0–5.5 V
pH	Analog pH Sensor/Meter Kit V2 (SEN0161-V2)	0–14 pH (±0.1 pH at 25°C)	Analog voltage; 3.3–5.5 V
DO	Analog Dissolved Oxygen Kit (SEN0237-A)	0–20 mg/L	Analog 0–3.0 V, galvanic BNC probe; 3.3–5.5 V
Turbidity	Analog Turbidity Sensor (SEN0189)	0–4000 NTU (response <500 ms)	Analog 0–4.5 V/digital; 5 V
EC	Analog EC Sensor K = 1 (DFR0300)	0–20 mS/cm (1–15 mS/cm recommended)	Analog 0–3.4 V, BNC probe; 3.0–5.0 V
TDS	Analog TDS Sensor/Meter (SEN0244)	0–1000 ppm (±10 %)	Analog; 3.3–5.5 V
ORP	Analog ORP Sensor/Meter (SEN0165)	±2000 mV	Analog, BNC probe; 5 V

Table II. Descriptive statistics of the analysis dataset ($n = 16{,}980$)

Parameter	Unit	Min	Max	Mean	Std	Median
Temperature	°C	23.25	32.75	25.23	0.93	25.12
pH	—	6.14	8.36	7.98	0.23	8.06
DO	mg/L	3.10	13.84	6.34	1.75	6.24
Turbidity	NTU	0.00	42.61	13.29	9.60	12.64
EC	µS/cm	40	2861	158.5	365.48	82
TDS	ppm	14	454	303.69	30.90	304
ORP	mV	49	799	519.83	146.50	458
WAWQI	—	8.60	94.85	71.30	11.04	71.39

Table III. Per-channel treatment of the cleaned record

Parameter	Bounds	Out of range	Spikes smoothed	Missing before	Interpolated	Still missing	Availability
Temperature (°C)	20.0–35.0	0	19	0	0	0	100.0 %
pH	4.0–10.0	1	40	1	1	0	100.0 %
DO (mg/L)	1.0–30.0	1	446	1	1	0	100.0 %
Turbidity (NTU)	0.0–200.0	0	819	0	0	0	100.0 %
EC (µS/cm)	20.0–3000.0	5249	495	5249	174	**5075**	**81.5 %**
TDS (ppm)	10.0–1000.0	108	38	108	32	76	99.7 %
ORP (mV)	−200.0–800.0	6	67	6	6	0	100.0 %

Counts are over the full cleaned record of 27,391 rows. Availability is the share of that record carrying a measured rather than an imputed value.
Bold entries mark the EC channel, whose probe failed during deployment.

Table IV. WAWQI constants, weights, and share of index variance

Parameter	Standard S_i	Ideal V_{ideal}	Weight (Brown)	Share (Brown)	Share (SHAP)
DO	5.0	14.6	0.4998	**67.7 %**	0.1 %
Turbidity	25.0	0.0	0.1000	**24.5 %**	17.1 %
pH	8.5	7.0	0.2940	**11.6 %**	0.3 %
TDS	300.0	0.0	0.0083	0.1 %	−1.8 %
ORP	300.0	0.0	0.0083	−0.4 %	−0.6 %
EC	400.0	0.0	0.0062	−0.5 %	**85.1 %**
Temperature	30.0	25.0	0.0833	−3.1 %	−0.3 %

Shares are covariance shares; negative values indicate a contribution moving opposite to the index. Each column sums to 100 % before rounding; the rounded entries sum to 99.9 %.
Bold entries indicate shares above 10 %.

2.08 at 10, so the plateau begins at 6, and below it the rule removes the short-term variation the forecasting task is meant to capture. Gaps of up to 10 rows are recovered by linear interpolation with forward fill limited to 10 rows; **backward fill is not used** because it propagates information from later observations into earlier ones. Gaps longer than 60 s are not bridged and start a new block.

Two channel-specific corrections apply. The EC probe failed during deployment, leaving 5,075 of 27,391 rows (18.5 %) with gaps too long to interpolate; these are filled with the channel median, and 1,302 rows (7.7 %) of the analysis subset carry that imputed value, which is used in Section IV.C when interpreting importance. The turbidity firmware set NTU to zero whenever sensor voltage fell in a 3.20–3.40 V band intended to detect a probe lifted from the water; the median submerged voltage is 3.382 V, inside that band, so valid readings were being zeroed. NTU is recomputed from voltage using the same firmware constants ($V_{clear} = 3.90$ V, sensitivity = 18.0 NTU/V) without that guard, and Section IV.D shows that the results are insensitive to both. The index is computed from the TDS probe reading rather than from a firmware-derived quantity equal to EC × 0.5, an available substitute whose consequence is quantified in Section III.C.

C. INDEX CONSTRUCTION

The WAWQI follows Brown *et al.* [1]. For each parameter i, the quality rating is $q_i = 100 \times |(V_i - V_{ideal,i})/(S_i - V_{ideal,i})|$, taking the absolute deviation so that a parameter is penalized for departing from its ideal in either direction, which matters for pH and temperature whose values fall on both sides. V_i is the measured value, S_i is the permissible standard [26], and $V_{ideal,i}$ is the value in pure water (14.6 mg/L for DO, 7.0 for pH, 0 otherwise). The index is $WAWQI = \Sigma w_i\, q_i / \Sigma w_i$ with $w_i = K/S_i$, normalized to sum to 1; reconstructing it from Table IV reproduces the index column of the analysis dataset at $r = 1.000000$.

The choice of weighting is not neutral, and two candidates are available for this dataset. Earlier work on the same site [8] addressed water quality as a classification problem, for which SHAP feature-importance values are a natural output, and those values (Turbidity 0.344, EC 0.311, TDS 0.185, DO 0.073, ORP 0.033, Temperature 0.027, and pH 0.027) offer an apparently data-driven set of index weights. For a forecasting task, the two behave differently: weights derived from a model's feature importances make the index a function of that model, so forecasting such an index and then asking which parameters matter returns the weighting decision rather than a property of the water body. The Brown formulation is independent of any model and is used here for that reason. Computed on identical rows the two indices correlate at $r = 0.265$, the SHAP-weighted index having mean 63.51 and standard deviation 30.12 against 71.30 and 11.04 for the Brown-weighted index; with the TDS slot additionally filled by EC × 0.5, EC carries 0.496 of unit weight across two slots, and those slots account for 91.2 % of the resulting index variance.

Table IV also shows why weights alone do not say what moves the index: a parameter's contribution depends jointly on its weight, its standard, and its own variability. Under the Brown weighting, the index is driven by DO and turbidity, and EC contributes −0.5 %; under the SHAP weighting, EC alone accounts for 85.1 %, being the most variable channel relative to its standard. On the standard classification scheme [1,2], the mean index of 71.30 places the lake in the *Poor* band (58.2 % *Poor*, 39.1 % *Very Poor*, 2.6 % *Good*, 0.1 % *Excellent*); none exceeds 100.

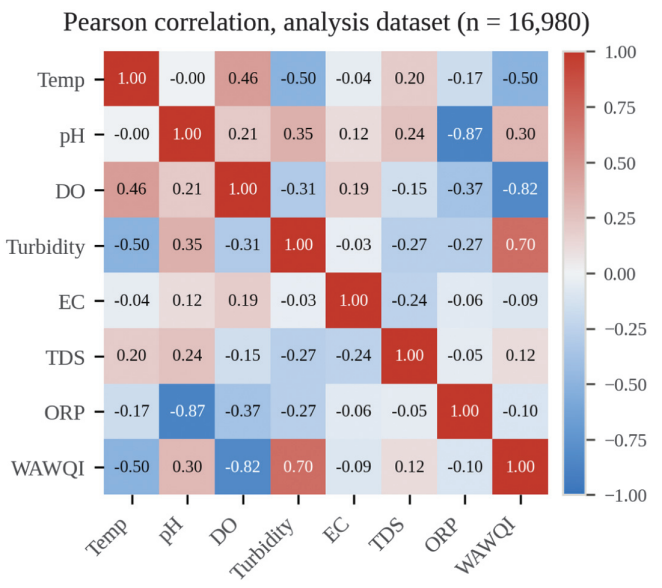


Fig. 4. Pearson's correlation between the seven sensor channels and the Brown-weighted index ($n = 16{,}980$). Diverging scale; the neutral midpoint is zero.

Figure 4 shows the pairwise correlations on the analysis dataset and gives the same ordering from a different direction. The index correlates −0.82 with DO, +0.70 with turbidity, and −0.50 with temperature, while its correlation with EC is −0.09. The negative sign on DO follows from the construction rather than from any inverse physical effect: a higher measured concentration lies closer to the ideal value of 14.6 mg/L, so its sub-index and therefore the composite score fall.

D. FORECASTING FORMULATION, FEATURES, AND HORIZONS

1). TARGET. The models predict the h-step change $\Delta_t(h) = y(t+h) - y(t)$, and the forecast is reconstructed as $\hat{y}(t+h) = y_t + \hat{\Delta}$. Predicting the level directly allows a model to score well by reproducing the current value, which is how a high coefficient of determination can coexist with no forecasting ability; predicting the change makes the comparison explicit because the persistence baseline is exactly $\hat{\Delta} = 0$. Both formulations are evaluated. The absolute target is run at the three shortest horizons (1, 5, and 15 min) as a comparator; across those 42 model–horizon–selection-rule combinations, it is significantly worse than persistence in 30 and better in none, whereas the change target is significantly worse in 14 of 70 and better in 1.

2). HORIZONS. Five horizons are used: 1, 5, 15, 30, and 60 min, that is, 4, 20, 60, 120, and 240 steps. They are chosen to span the decay of the index's own predictability rather than by convention: persistence attains $R^2 = 0.950$, 0.828, 0.654, 0.362, and −0.214 across them, and the index's own autocorrelation, computed within continuous blocks so that no pair spans a recording gap, decays from 0.981 to 0.969, 0.955, 0.931, and 0.868 over the same range. At 1 min, the index is nearly its own best forecast, leaving a model almost no room; at 60 min, the last observation is worse than the test-period mean, so the index's history has ceased to be useful. A comparison of methods is informative only between those points, and a set stopping at 15 min covers only the easier half. The endpoints also match operational scales: sensor-noise timescale at 1 min, and at 60 min the time an operator needs to respond to an oxygen deficit with manually operated aerators.

3). FEATURES. Eight features are constructed for each of eight base variables (the seven sensors and the index), giving 64 features: the contemporaneous value; lags of 1, 6, 11, and 16 steps (15 s, 1 min 30 s, 2 min 45 s, and 4 min); rolling mean and rolling standard deviation over a 20-step window (5 min), the same window at every horizon; and the first difference. Lags capture autoregressive structure, the rolling mean acts as a low-pass filter, the rolling standard deviation acts as a volatility indicator, and the first difference acts as the instantaneous rate of change [13,14,27]. The four lags are spaced five steps apart and span the same range as the rolling window, bracketing the shortest horizon of four steps; Section IV.C shows this choice is not critical. **Features are constructed strictly within a continuous block**, so no lag or

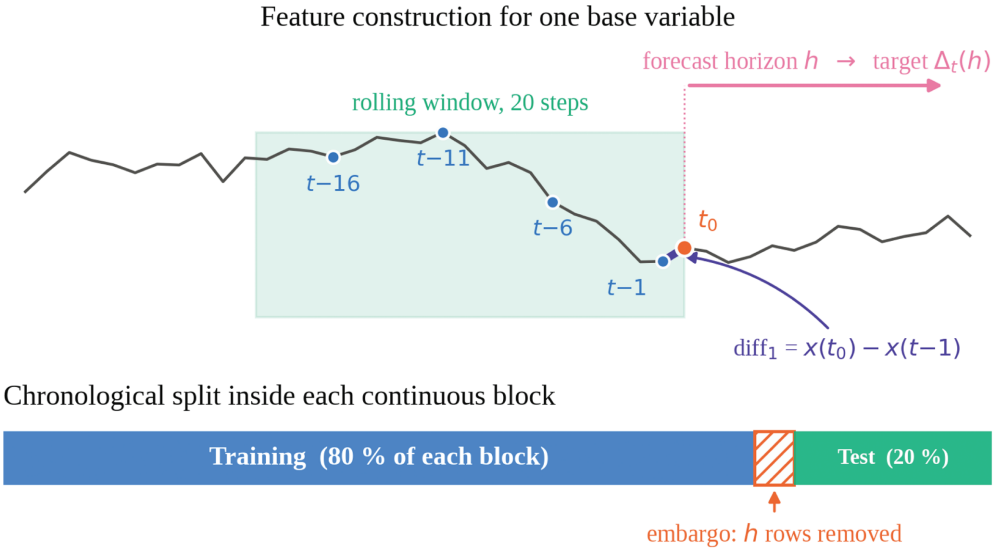


Fig. 5. Feature construction and leakage control. Upper panel: the features built from one base variable (one of the seven sensor channels or the index itself), showing the lags, the rolling window, the first difference, and the h-step target. Lower panel: chronological split within each continuous block, with an embargo of h rows at the test boundary.

rolling value is drawn across a recording gap or the power interruption. Figure 5 illustrates the construction for a single base variable together with the chronological split and the embargo described in Section III.E.

E. PROTOCOL, MODELS, AND EVALUATION

1). LEAKAGE CONTROL. An 80/20 chronological split is taken within blocks with no shuffling at any point. The h rows immediately preceding the test boundary are removed from training as an **embargo**; without it, a training row near the boundary carries a label drawn from the test period, a leak that produces no error and no warning [19]. Model selection uses TimeSeriesSplit (n_splits = 3, gap = h) computed entirely inside the training period [18], with any scaling fitted on the training fold alone, and no feature crosses a block boundary. **The test set is read exactly once**, after model selection is final.

2). MODELS AND OPTIMIZATION. Seven regressors are compared: Ridge [28], Random Forest [29], Extra Trees [30], Gradient Boosting [31], XGBoost [32], LightGBM [33], and CatBoost [34]. Each is tuned independently for every horizon and target using Optuna [17] with the TPE sampler (seed 42) and a median pruner at 50 trials per combination, 2,800 trials in total. **The objective is the mean Root Mean Squared Error (RMSE) over the `TimeSeriesSplit` folds, computed inside the training period only.** Table V gives the search spaces; the regularization ranges deliberately extend to values that force very simple models because narrower ranges cannot express a model simple enough for this data. Hyperparameters are selected by the **one-standard-error rule** [35]: among trials within one standard error of the best, the most regularized is chosen. At the 1-min horizon, the seven models' cross-validation scores span 5.7 %, while the standard error of a single estimate is approximately 0.35 in the same units, about three times that spread, so selecting the minimum of a set of tied scores selects the luckiest draw. Results under the conventional minimum-CV rule are reported alongside.

3). BASELINES. Persistence, $\hat{y}(t+h) = y_t$, is the primary reference. ARIMA [15] uses an order selected once by Akaike Information Criterion (AIC) on the 2,459 observations preceding the earliest test row of any horizon, never on test data, giving ARIMA(5,1,1); that order is then held fixed, while the model is refitted for each horizon on the 2,459 to 2,542 observations preceding that horizon's own test window, with forecasts produced by rolling one-step extension. LSTM [36] and GRU [37] use a random search over 10 configurations per horizon, 50 per architecture in total [38], with early stopping on a validation split held out from training. The search covers hidden units $\in \{16, 32, 64, 128\}$, layers $\in \{1, 2\}$, input window $\in \{5, 10, 20\}$, batch $\in \{128, 256\}$, dropout $\in \{0, 0.15, 0.30\}$, and learning rate $3 \times 10^{-4} - 5 \times 10^{-3}$. The input window is capped at 20 steps, so the recurrent models see the same sample set as the feature-based models, making the comparison one of method rather than of sample availability.

Table V. Hyperparameter search spaces

Model	Search space
Ridge	alpha 1e-3–1e5*
RF, Extra Trees	trees 100–500; depth 4–22; leaf 1–3000*; features {sqrt, log2, 0.5, 0.8}
Gradient Boosting	trees 100–400; lr 0.01–0.15*; depth 2–4; leaf 2–2000*; subsample 0.7–1.0
XGBoost	trees 200–700; lr 0.01–0.15*; depth 3–8; subsample and colsample 0.6–1.0; lambda 0.1–20*; child weight 1–2000*
LightGBM	trees 200–700; lr 0.01–0.1*; leaves 15–63; depth 3–8; child samples 5–3000*; subsample and colsample 0.6–1.0; lambda 0.1–30*; alpha 0.01–10*
CatBoost	iterations 200–600; lr 0.01–0.15*; depth 4–8; l2 1–300*; leaf 1–3000*

Ranges marked with an asterisk are sampled log-uniformly.

4). METRICS AND SIGNIFICANCE. RMSE, Mean Absolute Error (MAE), R^2, bias, and Mean Absolute Percentage Error (MAPE) are reported for every model, horizon, and target, with a 95 % confidence interval on RMSE from a block bootstrap (5,000 resamples, block length h), used instead of an ordinary bootstrap because forecast residuals are autocorrelated. The primary metric is the **skill score** $SS = 1 - MSE_{model}/MSE_{persistence}$, zero when a model adds nothing to the last observation; rigorous comparison against a naive forecast is necessary for series of low predictability [20]. Every comparison is tested with the **Diebold–Mariano test** [21] using a Bartlett HAC estimator at lag $h - 1$ with the Harvey–Leybourne–Newbold correction [22], alongside a paired block bootstrap on identical block indices. The study runs **167 tests**, at which point about eight nominally significant results would be expected by chance, so **Holm–Bonferroni** [23] and **Benjamini–Hochberg** [24] corrections are applied per family. The nine families are the four formed by crossing the two target formulations with the two selection rules, one each for ARIMA, LSTM, and GRU, one for the ablation, and one for the rolling-origin comparisons. Raw p and adjusted q are reported side by side, and *significant* is reserved for results surviving correction.

IV. RESULTS AND DISCUSSION

A. FORECAST ACCURACY AND SIGNIFICANCE

Because persistence attains $R^2 = 0.950$ at 1 min and -0.214 at 60 min, a model reporting $R^2 = 0.95$ at 1 min has demonstrated nothing beyond reproducing the last observation, while one reporting $R^2 = 0.36$ at 30 min may be an improvement on it. Reporting a persistence baseline is not by itself sufficient. At these horizons, the gaps between a model and the baseline are small relative to their sampling variability, so a numerically higher score can arise without any real improvement; every comparison reported here is therefore tested rather than read off the metric.

Table VI shows the skill score for all seven models at all five horizons, and Table VII shows the full error metrics for the highest-skill model at each horizon. Thirty-nine of the 70 model–horizon–selection-rule combinations achieve positive skill, but the magnitudes are small below 60 min. Figure 6 shows why: at 1 min, the observed series, the persistence baseline, and the model are visually indistinguishable, and as the horizon lengthens, both the baseline and the model increasingly lag the observed series rather than anticipating it, so the two error curves grow together.

MAPE is reported because it was requested and is flagged as uninformative here: the index sits near 71 in level, so MAPE varies only between 1.11 % and 7.46 % across the whole study, including for models whose skill is strongly negative. Of the 112 model–baseline comparisons in the main experiment, **one** is significant under both the Diebold–Mariano test and the paired block bootstrap: Random Forest at 5 min ($\Delta RMSE = -0.0199$, $p = 0.0026$), which survives Benjamini–Hochberg correction ($q = 0.018$) but not Holm–Bonferroni ($p_{_Holm} = 0.080$). Across all 167 tests, 64 reach

Table VI. Skill score against persistence, Δ target, 1-SE selection

Model	1 min	5 min	15 min	30 min	60 min
Ridge	−0.0013	+0.0091	+0.0160	+0.0250	+0.0599
Random Forest	+0.0044	+0.0158	+0.0084	+0.0138	+0.0544
Extra Trees	+0.0035	+0.0119	+0.0081	+0.0042	+0.0654
Gradient Boosting	−0.2273	−0.0371	+0.0052	−0.0223	+0.0934
XGBoost	−0.0790	+0.0303	−0.0303	−0.0250	+0.0514
LightGBM	−0.1886	+0.0364	−0.6431	−0.5451	+0.0724
CatBoost	−0.0055	−0.0021	−0.1093	+0.0472	−0.0105

Table VII. Error metrics for the highest-skill model at each horizon

Horizon	Model	RMSE	MAE	R^2	MAPE (%)	Skill	*p*
1 min	Random Forest	1.3458	0.8403	0.9503	1.11	+0.0044	0.507
5 min	LightGBM	2.4586	1.6582	0.8345	2.20	+0.0364	0.109
15 min	Ridge	3.5395	2.3527	0.6599	3.10	+0.0160	0.427
30 min	CatBoost	4.7562	3.6811	0.3923	4.93	+0.0472	0.550
60 min	Gradient Boosting	6.4387	5.3431	−0.1008	7.15	+0.0934	0.494

Persistence on the same rows gives RMSE 1.3487, 2.5046, 3.5683, 4.8726, and 6.7621; MAE 0.8412, 1.6160, 2.3462, 3.3570, and 5.1647; R^2 0.9501, 0.8283, 0.6544, 0.3622, and −0.2141; MAPE 1.11, 2.13, 3.10, 4.41, and 6.76 %. The complete table for all seven models, both target formulations and both selection rules, with bias and bootstrap intervals, is in the deposited archive. At 5 min, the highest-skill model is not the one whose improvement survives testing: Random Forest, at a lower skill of +0.0158, is the only significant improvement in the main experiment, so skill magnitude and statistical significance do not coincide here.

raw $p < 0.05$; correcting within each of the nine families, 51 survive Benjamini–Hochberg and 44 survive Holm–Bonferroni, and all but five of the survivors lie in the *worse than persistence* direction.

Figure 7 gives the distribution of absolute error at the 60-min horizon, where the models are furthest apart. Only Ridge has a median below the persistence median, 3.42 against 3.51 index points; the interquartile range of every model overlaps the baseline substantially, which is the distributional counterpart of the significance results reported above.

1). WHY THE LINEAR MODEL IS COMPETITIVE. Ridge attains the highest skill of the seven models at 15 min and lies within 0.006 of the highest at 1 min, so at those two horizons a regularized linear model holds its own against more flexible algorithms; at 5 min it ranks fifth, 0.027 below the highest. That has a mechanical explanation. The penalty selected for Ridge is 78,853 at 1 min and between 95,907 and 98,913 at the other horizons, against a search range bounded at 10^5. At penalties of that size, the coefficients shrink almost to zero, so the predicted change approaches the mean training change, itself close to zero: Ridge behaves as persistence plus a small mean-reversion term. It is competitive not because it models the series more accurately but because it models it very little, which on a target of this predictability is difficult to improve on. Since the selected penalty lies in the upper quarter of the search range at every horizon, the optimum may lie outside the range searched. The same reasoning explains why the more flexible algorithms fall furthest below the baseline when allowed to fit freely and why the selection rule matters: across the 56 model–horizon–target combinations, mean skill is −0.221 under the one-standard-error rule against −0.288 under minimum-CV selection, and the worst single outcome is −2.696 against −6.126.

2). DEGRADATION WITH HORIZON. The decline beyond 5 min is a property of the target rather than of the models: persistence R^2 falls from 0.828 to 0.654 between 5 and 15 min, while the standard deviation of the *h*-step change grows from 2.50 to 3.57 index points, and every model's RMSE grows with it; no model improves on persistence by more than 0.016 in skill at 15 min. Three results reported here bear on how longer horizons might be improved. Permutation importance concentrates increasingly on DO as the horizon lengthens, from 0.291 at 1 min to 0.796 at 60 min (Section IV.C), so the leverage lies in one channel's dynamics rather than a broader feature set. The ablation finds no measurable gain from the engineered feature families, so extending the feature set in the same direction is unlikely to help. And rolling-origin evaluation (Section IV.D) gives skill of +0.503 at 60 min in the one window with substantially elevated variability against approximately zero in the calmer ones, which suggests the limiting factor is how much of the record contains movement to forecast.

B. BASELINE COMPARISON

Table VIII compares ARIMA, LSTM, and GRU against the same persistence baseline on the same rows. ARIMA improves on persistence at raw significance at two horizons, but neither survives correction ($q = 0.065$ and 0.106) and the skill scores are +0.031 and +0.009; on this test window, ARIMA is not distinguishable from persistence. The recurrent models are worse at every horizon, and the mechanism is visible in the search log rather than the final scores: across the 50 trials run for each architecture, the best validation epoch is 4 on median for GRU and 5.5 for LSTM, and 7 and 20 for the five selected configurations, against a budget of 40 epochs with early stopping after five without improvement; validation loss therefore reaches its minimum well before the budget is exhausted in every case, which points to a model fitting structure that does not generalize rather than to an insufficient search. Simple methods matching or exceeding more complex sequence architectures on forecasting benchmarks is not specific to this dataset [25].

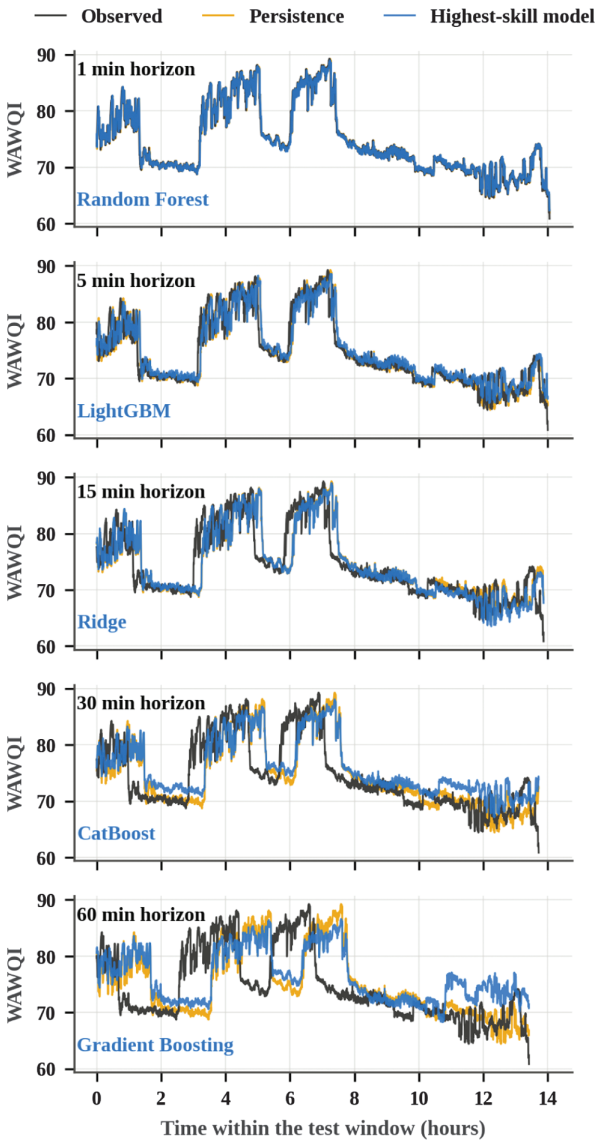


Fig. 6. Observed index, persistence baseline, and the highest-skill model over the test window, one panel per horizon. The three traces separate from the observed series together as the horizon lengthens.

Model complexity differs by orders of magnitude without a corresponding gain: the selected recurrent configurations carry up to 152,193 (GRU) and 202,881 (LSTM) parameters against six for the ARIMA(5,1,1) order selected here, and their architecture searches required 508 s and 373 s across the five horizons. What these costs mean for deployment is taken up in Section IV.E.



Fig. 7. Distribution of absolute error at the 60-min horizon under 1-SE selection. The dashed line marks the median absolute error of the persistence baseline on the same rows.

C. VARIABLE IMPORTANCE AND FEATURE CONTRIBUTION

Table IX reports the share of importance attributed to each variable by grouped permutation importance computed on the test set (10 repetitions) and by mean decrease in impurity computed during training. The two rank the variables differently, consistent with the documented bias of impurity-based importance toward variables with many distinct values [39]: turbidity, with 2,098 unique values, is ranked first by impurity at 1 min yet receives zero permutation importance, while temperature is ranked first by impurity at 60 min, its raw permutation effect is negative (−0.025), meaning the forecast improves when it is removed. Permutation importance is therefore the measure reported here, with impurity shown only for comparison.

1). EC DOES NOT DRIVE SHORT-TERM VARIATION OF THE BROWN-WEIGHTED INDEX. Under permutation importance, EC accounts for at most 0.002 of the total at any horizon. A result attributing short-term variation to EC is a property of the index it is computed on rather than of the water body: under the SHAP weighting, EC accounts for 85.1 % of index variance and 91.2 % once the TDS substitution is included (Section III.C), so a model forecasting that index must attend to it. Under the Brown weighting, EC contributes −0.5 % of index variance and 7.7 % of its rows carry an imputed median. What replaces it is DO. Since eight of the 64 features derive from the index itself, the analysis was repeated on the 56 sensor-only features: the ordering is unchanged and

Table VIII. Baseline forecasters against persistence

Horizon	ARIMA(5,1,1)	*p*	LSTM	*p*	GRU	*p*
1 min	−0.0202	0.286	−0.2231	0.000	−0.4038	0.000
5 min	+0.0307	0.013	−4.0897	0.000	−0.3464	0.000
15 min	+0.0094	0.238	−4.1482	0.000	−1.7340	0.000
30 min	+0.0089	0.042	−1.1269	0.023	−0.6894	0.029
60 min	+0.0067	0.087	−1.5859	0.033	−0.2491	0.518

Entries are skill scores against persistence; p is the raw Diebold–Mariano value.

Table IX. Variable importance shares, Extra Trees under the 1-SE rule

Variable	Perm. 1 min	Perm. 15 min	Perm. 60 min	MDI 1 min	MDI 15 min	MDI 60 min
WAWQI	0.415	0.577	0.194	0.280	0.184	0.066
DO	0.291	0.380	0.796	0.128	0.228	0.310
Temperature	0.255	0.000	0.000	0.059	0.318	0.430
Turbidity	0.000	0.034	0.000	0.380	0.047	0.016
ORP	0.036	0.008	0.010	0.049	0.124	0.098
pH	0.000	0.000	0.000	0.031	0.095	0.074
EC	0.001	0.000	0.000	0.047	0.002	0.005
TDS	0.002	0.000	0.000	0.025	0.001	0.000

Permutation shares are truncated at zero.

sharpens, with DO accounting for 0.709 at 1 min and 0.987 at 60 minutes, while removing the index-derived features costs little: skill falls from +0.0654 to +0.0513 at 60 min and by less than 0.008 elsewhere. The ranking is therefore not an artifact of including the target's own history.

Table X reports an ablation of the engineered feature families and gives a negative result that is reported as one. Of 20 comparisons against the full feature set, only one reaches raw significance, the removal of the first differences at 5 min ($p = 0.0083$, a skill difference of 0.0014), and it does not survive correction; at two horizons, the eight contemporaneous values give a higher skill score than the full set. The largest apparent difference, at 60 min, is not significant at this test-window size ($p = 0.384$). The claim is therefore narrow: on this record, at this sample size, no measurable contribution from the engineered feature families is demonstrated, which is a negative result under the present record length rather than evidence that such features are useless.

D. ROBUSTNESS

The corrected record spans approximately three days, so a design that trains on days 1–3 and tests on day 4 is unavailable; four consecutive equal test windows of 8.4–8.8 hours are used instead, and the comparison is separately repeated on the untruncated 27,271-observation record. Table XI places all six test subsets on one axis alongside the variability of the target in each.

Across the four rolling windows, the model separates from persistence only in the window whose target variability is roughly twice that of the others. That window is also where the recurrent baselines fail most severely (skill to −25.7) and where ARIMA could not be evaluated, its test rows not lying within a single continuous block, so no ARIMA figure exists for the window in which the ensemble performs best. Hyperparameters in these rows are held at the main-experiment values, tuned on the first 80 % of the record, so some windows fall inside that range and the choice is not fully out of sample even though each window's training is strictly causal; that bias favors the model.

The full-record row requires a different reading. Its test window lies entirely beyond the truncation point, in the period following the turbidity cleaning event, where the turbidity channel reads a mean of 4.38 NTU against 13.29 NTU in the analysis subset and the autocorrelation of the index at a lag of 240 steps, computed within continuous blocks as in Section III.D, falls from 0.868 to 0.287. Persistence derives its accuracy from that autocorrelation

Table X. Feature ablation, Extra Trees under the 1-SE rule

Configuration	Features	1 min	5 min	15 min	30 min	60 min
Full set	64	+0.0035	+0.0119	+0.0081	+0.0042	+0.0654
Without lags	32	+0.0031	+0.0110	+0.0085	−0.0000	+0.0522
Without rolling statistics	48	+0.0055	+0.0116	+0.0083	+0.0053	+0.0681
Without first differences	56	+0.0069	+0.0105	+0.0091	+0.0096	+0.0667
Contemporaneous values only	8	+0.0048	+0.0077	+0.0037	+0.0054	+0.0267

Entries are skill scores against persistence.

Table XI. Skill of Extra Trees under the 1-SE rule across test subsets

Test subset	σ	1 min	5 min	15 min	30 min	60 min
Rolling window 1	5.39	**+0.0225**	+0.0012	−0.0169	−0.0366	−0.0653
Rolling window 2	11.34	**+0.0105**	**+0.0207**	**+0.1216**	**+0.2088**	**+0.5026**
Rolling window 3	7.65	+0.0074	**+0.0124**	+0.0234	+0.0348	+0.0811
Rolling window 4	5.76	−0.0035	+0.0019	−0.0241	−0.0134	+0.0360
Main window, truncated record	6.04–6.14	+0.0035	+0.0119	+0.0081	+0.0042	+0.0654
Main window, full record	4.19–4.37	**+0.0189**	**+0.0360**	**+0.0748**	**+0.0768**	**+0.2941**

σ is the standard deviation of the target within the subset, quoted at the 1-min horizon for the rolling windows and as a range across horizons for the two main windows. Bold entries are significant at raw $p < 0.05$. Of the 20 rolling-origin comparisons, four survive Benjamini–Hochberg and three survive Holm–Bonferroni. The five full-record comparisons are not part of the 167-test correction family, and their p-values are raw.

and so degrades sharply there: normalized by the target standard deviation of its own test window, at 60 min the model's RMSE is essentially unchanged between the two records, at 1.065 and 1.070 times σ, while persistence rises from 1.102 to 1.273 times σ. The apparent improvement therefore measures deterioration of the baseline under altered measurement conditions rather than an improvement in the forecast. Truncation at the cleaning event keeps that effect out of the reported result.

Because turbidity accounts for 24.5 % of index variance and is partly reconstructed from voltage, both of its calibration constants were varied: clear-water voltage across 3.70–4.10 V and sensitivity across 14–22 NTU/V. Across all five variants and five horizons, skill ranges from +0.0030 to +0.0754, the largest deviation from the baseline configuration is 0.010, and no variant produces a significant improvement on persistence at any horizon.

Validation at a second site was not possible. The study rests on a single deployment, and no comparable high-frequency record from another aquaculture water body was available; what the rolling-origin and full-record rows establish therefore generalization across time within one site, not across sites. A second deployment with the same seven channels and an independently calibrated index remains the principal outstanding requirement before the protocol could be claimed to transfer.

E. PRACTICAL APPLICABILITY

Deployment feasibility rests on three questions: what a forecast costs to produce, how reliable the instrument that supplies it is, and whether the resulting forecast can support a warning. The first two are answered by measurements reported above; the third is not established by this study, and the bound on it is stated at the end of this section.

1). FORECAST COST. Inference is not the constraint at this sampling cadence. Per-sample inference times range from 0.0005 ms for Ridge to 0.048 ms for Random Forest, with the recurrent models at 0.012 ms (LSTM) and 0.016 ms (GRU); fitting a single model at a single horizon takes between 0.02 s (Ridge) and 40.6 s (Gradient Boosting), and the architecture search for the recurrent models required 373 s and 508 s across all five horizons. Against an acquisition interval of 15 s nominal and 16.00 s realized, the forecast step consumes a negligible fraction of one cycle, so end-to-end latency is set by the sensing cadence rather than by the model. On-node retraining is within reach for every model except Gradient Boosting. No measurement in this study supports the claim that a model of this class would be too slow to deploy.

2). INSTRUMENT RELIABILITY. The binding constraint is the instrument. Table III gives the per-channel outcome of preprocessing: availability, defined as the share of the cleaned record carrying a measured rather than an imputed value, is 100.0 % for temperature, pH, DO, and ORP, 99.7 % for TDS, and 81.5 % for EC, whose probe failed during deployment. A single power interruption divided the record into five continuous blocks and reset the real-time clock to its power-on default rather than stopping it, leaving wall-clock time for 10,984 of the 16,980 analysis rows (65 %) and well-formed but incorrect timestamps for the remainder.

What matters operationally is not that a channel failed but which one. The failed channel contributes −0.5 % of index variance under the Brown weighting and at most 0.002 of permutation importance at any horizon, so the loss of 18.5 % of its rows does not propagate to the forecast. Under the SHAP weighting, the same channel carries 85.1 % of index variance, and the identical hardware failure would have been disabling. Robustness to a probe failure is therefore a property of the index definition as much as of the hardware. Conversely, the channel that carries the index, DO at 67.7 % of variance and a permutation share rising to 0.796 at 60 min, had no unrecoverable rows but the second-largest count of smoothed spikes (446, 1.63 %), so noise handling matters most on the channel that matters most. Calibration uncertainty was tested rather than assumed: varying the turbidity constants across 3.70–4.10 V and 14–22 NTU/V moved skill by at most 0.010 at any horizon and never converted a non-significant result into a significant one. An early-warning deployment on this hardware would need redundancy on the DO channel and a clock independent of the node's power supply before it would need a faster model.

3). WARNING CAPABILITY. Whether such a system can warn is a separate question from whether it can forecast, and this study does not establish it. The results above bound what a warning would rest on. At 1 min, the last observation is already a near-perfect forecast (persistence $R^2 = 0.950$), so any alarm the model raises would be raised equally by the naive baseline. At 60 min, the model improves on persistence by the largest margin observed here, but the index's own history has by then ceased to be informative (persistence $R^2 = -0.214$), and that margin is confined to the single test window whose target variability is roughly twice that of the others (Table XI). Useful lead time and reliable anticipation therefore do not co-occur anywhere in this record. Establishing an operational warning capability would require a threshold-crossing evaluation on a record containing enough excursions to estimate both a detection rate and a false-alarm rate against a fixed index threshold, scored against persistence on identical rows. A single three-day deployment does not provide that, and reporting such rates from this record would repeat the error this study was designed to avoid: quoting a favorable statistic without the reference that makes it interpretable.

V. CONCLUSION

Short-term WAWQI forecasting was evaluated on 16,980 observations recorded at a nominal 15-s interval in an aquaculture lake, comparing seven Bayesian-optimized algorithms against persistence, ARIMA, LSTM, and GRU across five horizons under a leakage-controlled protocol with significance testing and multiple-testing correction.

The naive baseline is the decisive comparison at these horizons: persistence attains $R^2 = 0.950$ at 1 min and -0.214 at 60 min, so within that range a high coefficient of determination is a property of the target rather than evidence of forecasting ability. Of 112 model–baseline comparisons, one improves on persistence significantly under both tests: Random Forest under the one-standard-error rule at 5 min, surviving false-discovery-rate but not family-wise correction. Neither ARIMA nor the recurrent architectures improves on persistence after correction. Forecasting skill is conditional on how much the index moves: in rolling-origin evaluation, the ensemble reaches a skill score of 0.503 at 60 min in the window whose target variability is roughly twice that of the others, against approximately zero elsewhere. Under the Brown weighting, DO and turbidity account for 67.7 % and 24.5 % of index variance, while EC contributes −0.5 %, and permutation importance on held-out data ranks DO first among the sensor channels, its share rising to 0.796 at 60 min.

These results are bounded by a single site and a record of 16,980 observations spanning roughly three days, a failed

conductivity probe leaving 7.7 % of rows median-imputed, calibration constants taken from firmware and not independently verified, timestamps for 65 % of rows, and an ablation that does not establish a contribution from the engineered features at this sample size. The transferable result is the protocol: comparing an index-forecasting method against a naive baseline on identical rows, with an embargo, a single reading of the test set, significance testing and multiple-testing correction, changes the reported outcome substantially. Without those controls, the same models would appear to succeed.

DATA AVAILABILITY STATEMENT

The high-frequency time-series dataset generated and analyzed during the current study, together with the preprocessing, modeling, significance-testing, and robustness scripts that produce every table and figure reported here, is available in the Zenodo repository [40].

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Research and Community Service (PPM), Telkom University, and the Center for Excellence in Smart Technology and Applied Sciences (CoE STAS RG) for their support and facilitation in this work. Institutional support provided by Telkom University was instrumental in enabling the development, experimentation, and validation of the proposed Water Quality dataset framework. The study received funding from PPM with research grant number 6367/LIT06/PPM-LIT/2026.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] R. M. Brown *et al.*, "A water quality index—Do we dare?" *Water Sewage Works*, vol. 117, pp. 339–343, 1970.

[2] S. Tyagi *et al.*, "Water quality assessment in terms of water quality index," *Am. J. Water Resour.*, vol. 1, no. 3, pp. 34–38, 2013, DOI. 10.12691/ajwr-1-3-3.

[3] S. Chidiac *et al.*, "A comprehensive review of water quality indices (WQIs): History, models, attempts and perspectives," *Rev. Environ. Sci. Bio/Technol.*, vol. 22, no. 2, pp. 349–395, 2023, DOI. 10.1007/s11157-023-09650-7.

[4] R. Baena-Navarro *et al.*, "Intelligent prediction and continuous monitoring of water quality in aquaculture: Integration of machine learning and internet of things for sustainable management," *Water*, vol. 17, no. 1, art. no. 82, Jan. 2025, DOI. 10.3390/w17010082.

[5] K. Lal *et al.*, "Low-cost IoT based system for lake water quality monitoring," *PLoS One*, vol. 19, no. 3, art. no. e0299089, Mar. 2024, DOI. 10.1371/journal.pone.0299089.

[6] U. Ahmed *et al.*, "Efficient water quality prediction using supervised machine learning," *Water*, vol. 11, no. 11, art. no. 2210, 2019, DOI. 10.3390/w11112210.

[7] R. Choudhary *et al.*, "Predicting water quality index using stacked ensemble regression and SHAP-based explainable artificial intelligence," *Sci. Rep.*, vol. 15, art. no. 31139, 2025, DOI. 10.1038/s41598-025-09463-4.

[8] G. A. Mutiara, M. R. Alfarisi, and L. Meisaroh, "Smart multisensor-based machine learning for multiclass freshwater fishpond water quality degradation classification," *Eng. Technol. Appl. Sci. Res.*, vol. 16, no. 3, pp. 34795–34806, 2026, DOI. 10.48084/etasr.17840.

[9] J. D. Willard *et al.*, "Time series predictions in unmonitored sites: A survey of machine learning techniques in water resources," *Environ. Data Sci.*, vol. 4, art. no. e7, 2025, DOI. 10.1017/eds.2024.14.

[10] A. W. Jayawardena and F. Lai, "Analysis and prediction of chaos in rainfall and stream flow time series," *J. Hydrol.*, vol. 153, no. 1–4, pp. 23–52, 1994, DOI. 10.1016/0022-1694(94)90185-6.

[11] B. Sivakumar, "Chaos theory in hydrology: Important issues and interpretations," *J. Hydrol.*, vol. 227, no. 1–4, pp. 1–20, 2000, DOI. 10.1016/S0022-1694(99)00186-9.

[12] C. T. Doan and H. D. Nguyen, "Robust water quality prediction across multiple indicator formulations using an explainable ensemble learning model," *Water Resour. Ind.*, vol. 34, art. no. 100329, 2025, DOI. 10.1016/j.wri.2025.100329.

[13] Y. Safat *et al.*, "Feature-enhanced rainfall prediction in Bangladesh using statistical and machine learning models integrated with explainable AI," in *Proc. 2nd Int. Conf. on Next-Generation Computing, IoT and Machine Learning (NCIM)*, Gazipur, Bangladesh, 2025, pp. 1–6, DOI. 10.1109/NCIM65934.2025.11159999.

[14] J. M. Lorente-González *et al.*, "Machine learning techniques for modelling the water quality of Coastal Lagoons," *Water*, vol. 18, no. 3, art. no. 297, Jan. 2026, DOI. 10.3390/w18030297.

[15] J. Su *et al.*, "A hybrid model of ARIMA and MLP with a Grasshopper optimization algorithm for time series forecasting of water quality," *Sci. Rep.*, vol. 14, art. no. 23927, 2024, DOI. 10.1038/s41598-024-74144-7.

[16] L. Yang and A. Shami, "On hyperparameter optimization of Machine Learning Algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, DOI. 10.1016/j.neucom.2020.07.061.

[17] T. Akiba *et al.*, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Anchorage, AK, USA, 2019, pp. 2623–2631, DOI. 10.1145/3292500.3330701.

[18] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci.*, vol. 191, pp. 192–213, 2012, DOI. 10.1016/j.ins.2011.12.028.

[19] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns (N Y)*, vol. 4, no. 9, art. no. 100804, 2023, DOI. 10.1016/j.patter.2023.100804.

[20] N. Beck, J. Dovern, and S. Vogl, "Mind the naive forecast! A rigorous evaluation of forecasting models for time series with low predictability," *Appl. Intell.*, vol. 55, no. 6, art. no. 395, 2025, DOI. 10.1007/s10489-025-06268-w.

[21] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *J. Bus. Econ. Stat.*, vol. 13, no. 3, pp. 253–263, 1995, DOI. 10.1080/07350015.1995.10524599.

[22] D. Harvey, S. Leybourne, and P. Newbold, "Testing the equality of prediction mean squared errors," *Int. J. Forecast.*, vol. 13, no. 2, pp. 281–291, 1997, DOI. 10.1016/S0169-2070(96)00719-4.

[23] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Stat.*, vol. 6, no. 2, pp. 65–70, 1979.

[24] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. R Stat. Soc.: Series B*, vol. 57, no. 1, pp. 289–300, 1995, DOI. 10.1111/j.2517-6161.1995.tb02031.x.

[25] A. Zeng *et al.*, "Are transformers effective for time series forecasting?," in *Proc. AAAI Conf. on Artif. Intell.*, vol. 37, no. 9, 2023, pp. 11121–11128, DOI. 10.1609/aaai.v37i9.26317.

[26] World Health Organization, Guidelines for Drinking-Water Quality, 4th edn., 1st addendum. Geneva: WHO Press, 2022. [Online] Available: https://www.who.int/publications/i/item/9789240045064

[27] L. Fazeldehkordi and J.-L. Chiang, "The impact of window size on the performance and accuracy of time series forecasting models for meteorological drought prediction," *Stochastic Environ. Res. Risk Assess.*, vol. 39, no. 12, pp. 6155–6180, 2025, DOI. 10.1007/s00477-025-03098-7.

[28] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, DOI. 10.1080/00401706.1970.10488634.

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, DOI. 10.1023/A:1010933404324.

[30] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006, DOI. 10.1007/s10994-006-6226-1.

[31] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001, DOI. 10.1214/aos/1013203451.

[32] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794, DOI. 10.1145/2939672.2939785.

[33] G. Ke *et al.*, "LightGBM: A highly efficient gradient Boosting Decision Tree," in *Proc. 31st Conf. on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 3146–3154.

[34] L. Prokhorenkova *et al.*, "CatBoost: Unbiased boosting with categorical features," in *Proc. 32nd Conf. on Neural Information Processing Systems (NeurIPS)*, Montréal, Canada, 2018, pp. 6638–6648.

[35] T. Hastie, R. Tibshirani, and J. Friedman,The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009, sec. 7.10.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734, DOI. 10.3115/v1/D14-1179.

[38] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res*, vol. 13, pp. 281–305, 2012.

[39] C. Strobl *et al.*, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinf.*, vol. 8, art. no. 25, 2007, DOI. 10.1186/1471-2105-8-25.

[40] I. F. Putri, G. A. Mutiara, and M. R. Alfarisi, "High-frequency water quality time-series dataset for WAWQI forecasting," version 2, Zenodo, 2026, DOI. 10.5281/zenodo.22750751.