Data-Audited Machine Learning for Reliable Water Quality Prediction and Classification in Smallholder Aquaculture

Nur Fatihah binti Azmi[1]*, Muhammad Alif Aiman bin Mohd Fauzi[2], Ammar Yasir bin Abdul Jalil[3], Mas Haslinda binti Mohamad[4]

[1,2,4]Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia

[3] Universiti Kuala Lumpur (UniKL) Malaysia-France Institute (MFI), Japan Collaboration Program Section, Jalan Damai, Seksyen 14, 43650 Bandar Baru Bangi, Selangor, Malaysia

***Corresponding author**

DOI: https://doi.org/10.47772/IJRISS.2026.101000152

Received: 12 October 2026; Accepted: 17 October 2026; Published: 30 October 2026

ABSTRACT

Stable water quality is essential for aquaculture productivity, but Internet of Things (IoT) sensor records from small farms may contain faulty readings that reduce the reliability of data-driven decisions. This study developed and evaluated a data-audited machine-learning decision-support pipeline for smallholder aquaculture water-quality monitoring. The workflow combined site-level sensor screening, pH estimation, feature-importance interpretation, water-status classification, dashboard visualisation, and Telegram alerting. Historical IoT records from a tilapia farm were screened using physical-validity rules before model development. Sites 1 and 2 were retained for modelling, while Site 5 was restricted to the audit because 195 of its 211 pH readings were physically invalid. The final regression and classification datasets contained 126 and 174 samples, respectively. An 80:20 split was used for both tasks, with stratification applied to classification. The Random Forest Regressor estimated pH from temperature, turbidity, and total dissolved solids with R-squared = 0.9337, RMSE = 0.0859, and MAE = 0.0471. Mean-decrease-in-impurity importance ranked TDS first at 64.50%, followed by temperature at 20.36% and turbidity at 15.15%. The Random Forest Classifier achieved 97.14% accuracy and 94.12% Critical-condition recall, but one of 17 Critical samples was misclassified, giving a 5.88% false-negative rate. Because the class labels were generated from fixed thresholds, the rule-only baseline reproduced those labels by construction and remained the authoritative safety layer. The dashboard therefore used machine learning as supplementary decision support while retaining a deterministic override for unsafe conditions in data-limited farm settings. The findings support an audited, interpretable, and conservative deployment strategy rather than reliance on predictive accuracy alone.

Keywords: Aquaculture 4.0, Data quality audit, Random forest, Sensor reliability, Water quality prediction

INTRODUCTION

Fish health can deteriorate quickly when pond conditions move outside a safe range. pH affects the chemical balance of the water, temperature changes fish metabolism, turbidity represents suspended material, and TDS captures dissolved minerals, salts, and waste [1], [2]. These variables also interact. A shift in one measurement may therefore change how the others should be interpreted, especially during feeding, rainfall, or water exchange.

Continuous sensing does not remove the need to inspect the data. Low-cost probes drift, lose calibration, or stop reporting correctly. Their outputs may still look numerical and enter a database without raising an obvious warning. If those records are used for training, the model learns from sensor faults rather than pond behaviour [9], [10]. Sensor reliability was therefore treated as the first engineering problem in the study. Every record passed through a physical-validity and site-level audit before model fitting.

Farm operators need more than a screen of raw readings. Decision-support systems can translate water-quality measurements into information that supports aquaculture management decisions [8]. This is difficult because water-quality relationships are non-linear [3], [4]. Feature-importance analysis can help identify the variables that contribute to water-quality predictions and support model interpretation [5], [16], [17], [18]. Both functions depend on trustworthy input data.

The study had three measurable objectives Published systems usually address only part of this workflow. IoT studies often concentrate on sensing and transmission [1], [2], [25]. Previous studies have investigated machine-learning approaches for predicting water-quality parameters and indices [3], [4], [12], [13], [14], [20]. Other studies have examined prediction under missing-data conditions [15] and long-term urban-lake monitoring [19]. Classification studies use water-quality measurements to distinguish suitability classes or water-quality conditions in general water-assessment and aquaculture applications [6], [7], [10], [11], [22], [23], [24]. Random Forest is already a common baseline in this field. The engineering contribution lies in the sequence around the model: site-level data auditing, pH-source substitution when the sensor becomes doubtful, and a fixed safety rule that can overrule the classifier. The same pipeline carries the result to a farm-facing dashboard.

. The first was to estimate pH from temperature, turbidity, and TDS with R-squared above 0.90 and RMSE below 0.10. The second was to interpret the fitted model by ranking the influence of the three input variables. The third was to classify water quality as optimal or critical with at least 85% accuracy and to integrate the output into a dashboard with alert notification. These objectives position the contribution as an audited decision-support pipeline rather than a fully automated aquaculture control system.

The central contribution of the paper is therefore not the use of Random Forest alone, but its placement within a data-audited decision-support pipeline. The workflow links site-level sensor screening, pH estimation, feature-importance interpretation, threshold-based safety override, dashboard visualisation, and Telegram alerting. This integration translates model outputs into practical operational information while retaining deterministic safeguards for safety-critical decisions.

METHODOLOGY

Figure 1 illustrates the workflow of the data-audited machine-learning pipeline, comprising data acquisition, site-level auditing, preprocessing, regression modelling, water-status classification, dashboard integration, and Telegram alerting. Sensor records are merged by timestamp and audited for physical validity, breaches of operating limits, missing values, and site suitability for model development. Separate cleaned datasets are prepared for pH regression and water-status classification to accommodate their respective target definitions and filtering criteria.

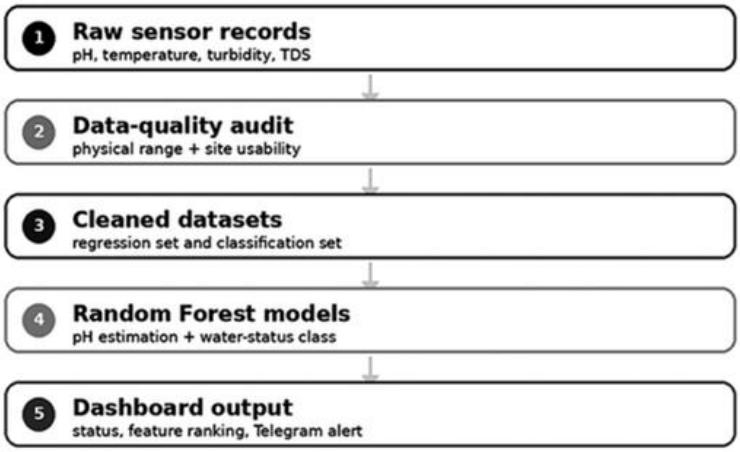


Figure 1. Workflow of the data-audited machine-learning pipeline for aquaculture water-quality decision support.

Dataset And Data-Quality Assessment

The study used historical IoT water-quality records collected from five aquaculture monitoring sites. Each monitoring site stored pH, temperature, turbidity, and TDS readings in separate CSV files, which were

standardized and joined using the timestamp field. The initial audit covered 587 usable timestamped samples: 188 from Site 1, 188 from Site 2, and 211 from Site 5. Sites 3 and 4 were removed before model development because too few usable readings remained after sensor-related faults. Site 5 was retained only for the data-quality audit because 195 of its 211 pH records were outside the valid physical pH range. Consequently, only Sites 1 and 2 were used for model training and testing. This decision reduced site coverage but prevented severe pH sensor faults from contaminating the machine-learning models.

Table 1. Site-level exclusion rationale used to protect model training from severe sensor faults

Site	Raw samples	Main data-quality issue	Use in paper
Site 1	188	Usable after cleaning	Model training and testing
Site 2	188	Usable after cleaning	Model training and testing
Site 3	Not retained	Too few usable records after sensor faults	Excluded
Site 4	Not retained	Too few usable records after sensor faults	Excluded
Site 5	211	195 pH readings outside valid physical range	Audit only

Preprocessing And Feature Scaling

Cleaning was performed in several passes. The combined Site 1 and Site 2 training file contained 376 samples. The first filter retained only physically valid values: pH greater than 0 and no greater than 14, temperature greater than 0 and below 50 °C, TDS greater than 0 ppm, and turbidity of at least 0 NTU. This general boundary filtering produced 223 cleaned samples. For regression, an additional operating-range filter retained pH values from 6.0 to 8.5, producing 126 samples. For classification, an additional screening step retained records with pH values between 4.0 and 14.0, reducing the physically filtered dataset from 223 to 174 samples. The retained samples were then assigned threshold-derived Optimal or Critical labels. Separate MinMaxScaler objects were used for the two tasks. Each scaler was fitted exclusively to its training subset and then applied to the corresponding test subset, preventing test-set information from entering the preprocessing stage. The fitted scalers were serialised with the models and reused by the dashboard to preserve deployment consistency. Code Listing 1. Illustrative excerpt of the data-auditing, boundary-filtering, and water-status-labelling procedure.

Code Listing 1 presents an illustrative excerpt rather than a standalone executable script; dataset preparation and column-variable definitions are omitted for brevity.

```
training_df = master_df[master_df["Site_ID"].isin(["Site_1", "Site_2"])]
clean_df = training_df[(training_df["pH"] > 0) & (training_df["pH"] <=
14) & (training_df["Temperature"] > 0) & (training_df["Temperature"] <
50) & (training_df["TDS"] > 0) & (training_df["Turbidity"] >= 0)]
regression_df = clean_df[clean_df["pH"].between(6.0, 8.5)]
classification_df["Status"] = np.where((pH.between(6.5, 8.5)) &
(Temperature.between(25, 32)) & (Turbidity < 30) & (TDS <= 500),
"Optimal", "Critical")
```

pH Estimation And Model Interpretation

Temperature, turbidity, and TDS were used as predictors for pH estimation. The 126 regression samples were divided into 80% training data and 20% test data using random_state = 71 with shuffling enabled. Random Forest combines multiple decision trees to perform regression or classification [21]. A Random Forest Regressor was trained using 800 trees, random_state = 42, n_jobs = -1, min_samples_leaf = 1, min_samples_split = 2, and max_depth = None. Model performance was evaluated using R-squared, root mean squared error (RMSE), and mean absolute error (MAE). The predefined success criteria were R-squared above 0.90 and RMSE below 0.10.

Mean decrease in impurity (MDI) feature importance was used to rank the relative contribution of temperature, turbidity, and TDS to pH prediction.

The pH estimation model and regression performance metrics were expressed as follows:

$$pH = f(T, Tu, TDS) \quad \text{Eq. 1}$$

$$R^2 = 1 - \left[\frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2}\right] \quad \text{Eq. 2}$$

$$RMSE = \sqrt{\left[\left(\frac{1}{n}\right)\Sigma(y_i - \hat{y}_i)^2\right]} \quad \text{Eq. 3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad \text{Eq. 4}$$

where T is temperature, Tu is turbidity, TDS is total dissolved solids, y_i is the measured pH, $\hat{y}_i$ is the predicted pH, $\bar{y}$ is the mean measured pH, and n is the number of test samples.

Water-Status Classification Model

The classification model used pH, temperature, turbidity, and TDS as input variables. A sample was labelled Optimal only when all four parameters satisfied the predefined operating limits: pH of 6.5-8.5, temperature of 25-32 °C, turbidity below 30 NTU, and TDS not exceeding 500 ppm. Any breach produced a Critical label. The 174 labelled samples were divided using an 80:20 stratified split with random_state = 42 so that both subsets retained similar class proportions. The Random Forest Classifier used 300 trees, random_state = 42, n_jobs = -1, and class_weight = balanced. Performance was evaluated using accuracy, precision, recall, F1-score, support, and confusion-matrix analysis. Critical-condition recall and the false-negative rate were emphasised because an unsafe condition classified as Optimal may delay corrective action.

The classification performance metrics were calculated using:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. 5}$$

$$Precision = \frac{TP}{TP + FP} \quad \text{Eq. 6}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{Eq. 7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \text{Eq. 8}$$

For the safety-focused evaluation, Critical was treated as the positive class. TP denotes Critical samples correctly classified as Critical; TN denotes Optimal samples correctly classified as Optimal; FP denotes Optimal samples classified as Critical; and FN denotes Critical samples classified as Optimal. The Critical false-negative rate was calculated as $FN/(TP + FN)$. These metrics were interpreted together because a high overall accuracy can still hide a safety-relevant false negative when a critical water condition is classified as optimal.

Deterministic Baseline And Safety-Layer Evaluation

A deterministic rule-only baseline was evaluated on the held-out classification samples using the same four thresholds that generated the reference labels: pH outside 6.5–8.5, temperature outside 25–32 °C, turbidity at least 30 NTU, or TDS above 500 ppm. Because the reference labels were derived from these rules, the baseline reproduced them by construction and did not constitute independent predictive evidence. The comparison reported overall accuracy, Critical-condition recall, and the Critical false-negative rate for the Random Forest Classifier and the deterministic baseline. This label-based comparison was distinct from the dashboard override implementation, which used different temperature and turbidity boundaries.

Dashboard Deployment And Alert Logic

The trained Random Forest Regressor and Random Forest Classifier were integrated into a Streamlit-based dashboard to translate model outputs into operational decision-support information. The dashboard displays actual sensor pH, temperature, turbidity, TDS, AI-estimated pH, the pH source used for classification, and the final water-quality status. Under normal conditions, the actual sensor pH is used for classification because it represents the direct measurement of pond water. However, when the pH reading is invalid or differs substantially from the AI-estimated pH, the system can use the AI-estimated pH as a fallback input. In this study, a pH discrepancy greater than 1.0 was treated as a potential sensor anomaly. Both the classifier and the dashboard override use the selected pH value, which is determined by these sensor-validity and discrepancy checks and may therefore differ from the original sensor reading.

A rule-based safety override was embedded in the dashboard to flag specified threshold breaches even when the classifier returned an Optimal label. In the implementation documented in the thesis, the status was forced to Critical when the selected pH was below 6.5 or above 8.5, temperature was below 25 °C or above 35 °C, turbidity exceeded 30 NTU, or TDS exceeded 500 ppm. The upper temperature limit and turbidity boundary therefore differed from those used to generate the classification labels. When a Critical condition was detected, the system sent a Telegram alert containing the input readings, AI-estimated pH, selected pH source, final status, and reason for the alert.

Code Listing 2 presents an abbreviated training and serialisation excerpt. The scaler-serialisation steps are omitted from the listing and the complete implementation saves the regression and classification scalers alongside their respective model.

Code Listing 2. Random Forest training, interpretation output, and model serialisation for dashboard deployment.

```
rfr = RandomForestRegressor(n_estimators=800, random_state=42, n_jobs=-
1, min_samples_leaf=1, min_samples_split=2, max_depth=None)
rfr.fit(X_train_reg, y_train_reg)
y_pred = rfr.predict(X_test_reg)
feature_importance = pd.Series(rfr.feature_importances_,
index=["Temperature", "Turbidity", "TDS"])
rfc = RandomForestClassifier(n_estimators=300, random_state=42, n_jobs=-
1, class_weight="balanced")
rfc.fit(X_train_cls, y_train_cls)
joblib.dump(rfr, "rf_regressor_ph.pkl")
joblib.dump(rfc, "rf_classifier_status.pkl")
```

RESULTS AND DISCUSSION

Data Cleaning And Dataset Preparation

pH failure caused most of the data loss in the raw audit. Among 587 raw samples, 497 breached at least one water-quality rule and 348 contained a sensor error. Unsafe pH occurred in 339 samples, while 212 pH readings fell outside the physical range. Temperature accounted for 170 unsafe and 136 invalid values, and invalid TDS occurred 136 times. The audit found no missing values and no turbidity faults. These counts overlap, so their sum is larger than the number of samples. The scale of the pH failure explains why site screening had to precede model training.

Table 2. Site-Level Audit Retained Sites 1 And 2 For Modelling And Restricted Site 5 To Data-Quality Analysis

Site	Raw samples	Use in this study
Site 1	188	Model training after cleaning
Site 2	188	Model training after cleaning
Site 5	211	Data-quality audit only
Total	587	Raw combined set before cleaning

Table 3. pH Faults Dominated The 587-Sample Pre-Cleaning Problem Profile

Problem type	Samples
Has water-quality problem	497
Has sensor error	348
Unsafe pH	339
Invalid pH	212
Unsafe temperature	170
Invalid temperature	136
Invalid TDS	136
Unsafe TDS	8
Missing values (any)	0
Turbidity faults (any)	0

Site selection and physical filtering removed most of the unusable records before either model was trained. Sites 1 and 2 contributed 376 raw training samples after Site 5 was excluded. Boundary filtering retained 223 samples, equal to 59.3% of the two-site training set. The regression-specific pH restriction produced 126 samples, whereas additional pH screening between 4.0 and 14.0 retained 174 samples for classification. Table 4 records this attrition. The different final counts reflect the purpose of each task: regression required a narrower pH range for stable estimation, while classification retained a wider range to represent critical conditions.

Table 4. Data Auditing Reduced 587 Raw Samples To 126 Regression And 174 Classification Samples

Stage	Samples
Raw combined set before cleaning	587
Raw training set (Site 1 + Site 2)	376
General cleaned set after boundary filtering	223
Regression set (pH 6.0-8.5)	126
Classification set (optimal/critical)	174

Predictive Performance For pH Estimation

The pH estimator met both predefined performance targets on the held-out test set. The Random Forest Regressor

achieved R-squared of 0.9337, RMSE of 0.0859, and MAE of 0.0471 (Table 5). On the held-out test set, the model explained approximately 93.4% of the variance in measured pH. The small gap between RMSE and MAE suggests that the test result was not dominated by a few unusually large errors. The result is still bounded by the experimental design: the estimate comes from one split of 126 samples collected from two sites after restricting pH to 6.0-8.5. It shows fit within the cleaned operating range, not transferability to another farm, season, or sensor configuration.

Table 5. The pH Regressor Met The R-Squared And RMSE Targets On The Held-Out Test Set

Metric	Result	Target	Status
R-squared	0.9337	> 0.90	Achieved
RMSE	0.0859	< 0.10	Achieved
MAE	0.0471	-	Reported

Environmental Interpretation Of Predictor Influence

TDS was the strongest predictor of pH in the fitted forest. Its MDI share was 64.50%, compared with 20.36% for temperature and 15.15% for turbidity (Table 6; Figure 2). This ordering is plausible for the observed ponds because dissolved minerals, salts, feed residue, and fish waste affect TDS and may coincide with chemical conditions associated with pH. Temperature ranked second, which is consistent with its influence on biological activity and decomposition. Turbidity had a smaller role in pH prediction within this dataset, although it remains relevant to pond management. The ranking reflects the relative contribution of each predictor within the fitted model and should not be interpreted as causal evidence. MDI importance may favour variables that offer more candidate split points. Future evaluation should assess the stability of this ranking using permutation importance or SHAP analysis.

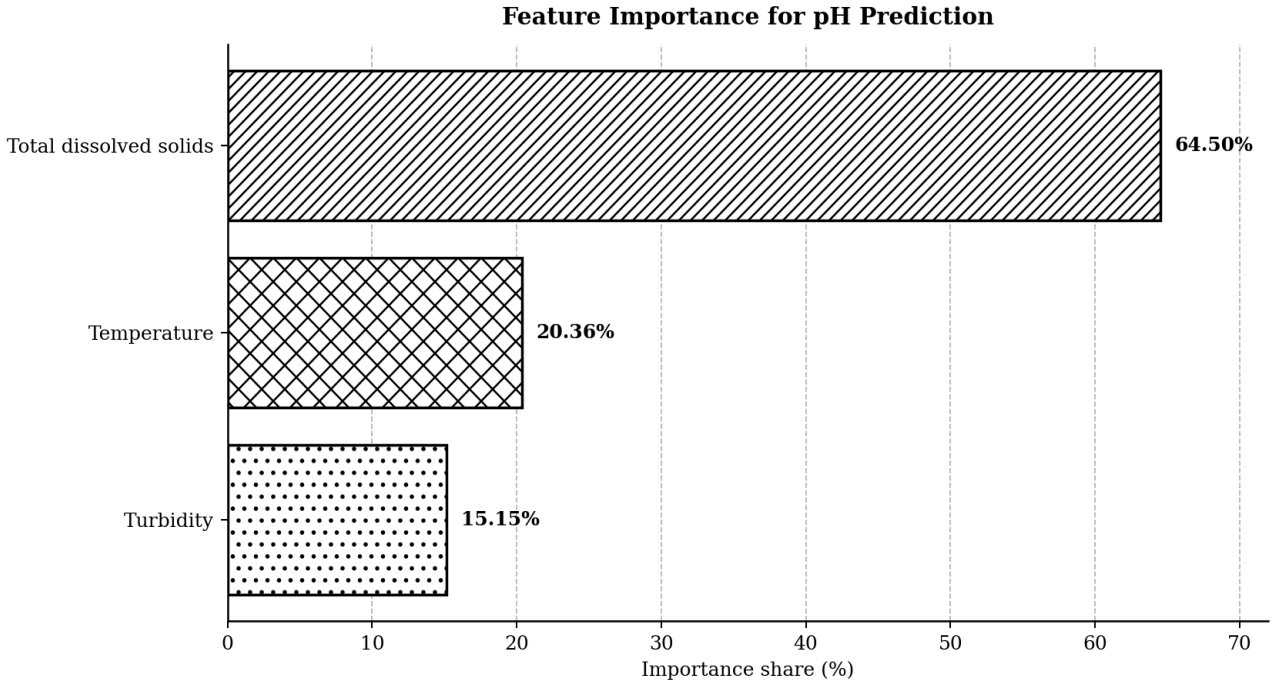


Figure 2. TDS contributed more than three times the MDI importance of temperature in the pH model

Table 6. TDS Contributed 64.50% Of The Model's MDI Importance For pH Prediction

Rank	Feature	Importance score	Share (%)
1	Total dissolved solids	0.644966	64.50
2	Temperature	0.203583	20.36
3	Turbidity	0.151451	15.15

Operational Water-Status Classification

Water-status classification exceeded the 85% accuracy target but produced one safety-relevant error. The classifier labelled 34 of 35 test samples correctly and reached 97.14% accuracy. All 18 Optimal samples were correctly identified. Among 17 Critical samples, 16 were detected and one was labelled Optimal (Table 7; Figure 3). Critical-condition recall was therefore 94.12%, and the Critical false-negative rate was 5.88%. The test set is too small to attribute the error confidently to a particular parameter boundary. The result nevertheless shows why overall accuracy must be interpreted together with safety-specific errors.

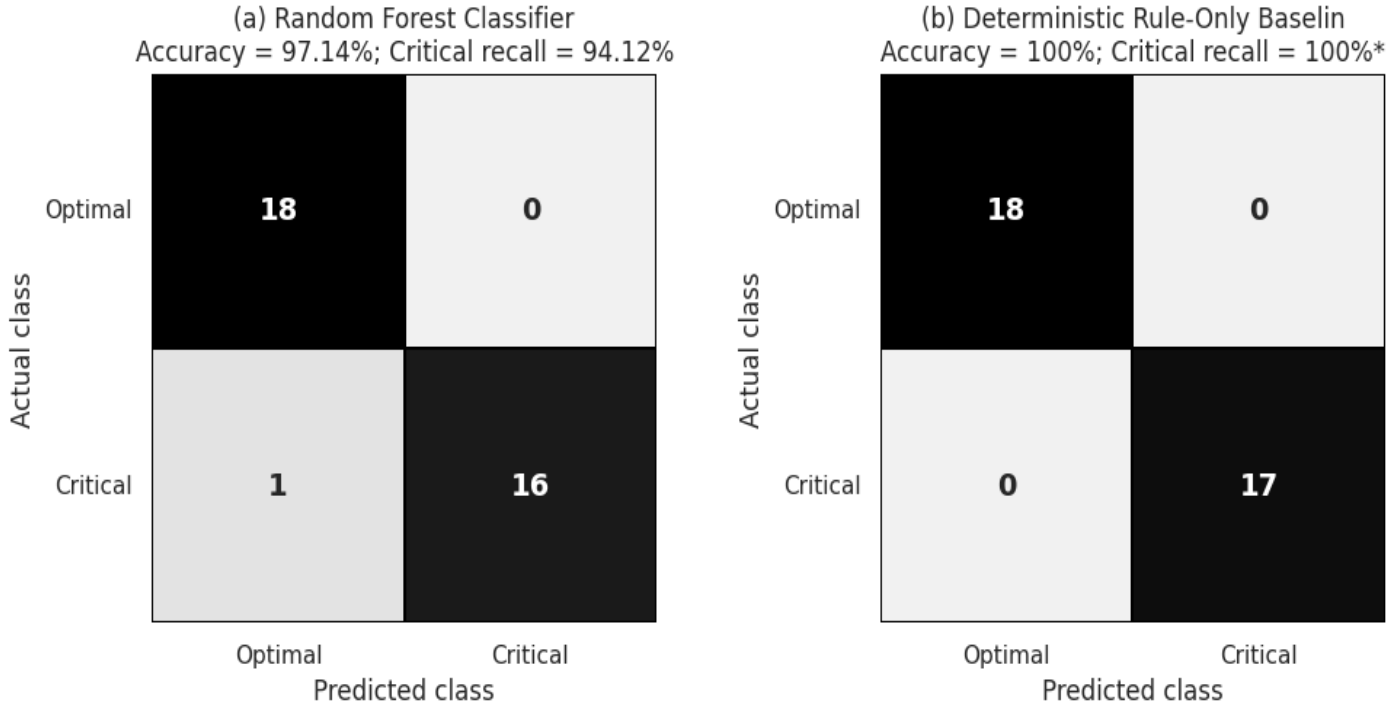


Figure 3. Confusion matrices for (a) the Random Forest Classifier and (b) the deterministic rule-only baseline. The baseline reproduces the reference labels by construction because both use the same thresholds.

The deterministic rule-only baseline reproduced all 35 reference labels. This agreement was expected because the baseline applied the same thresholds used to generate the labels. The result represents a label-consistency comparison rather than independent predictive performance or a verified evaluation of the dashboard override.

Table 7. Classification performance reached 97.14% accuracy with one critical false negative

Class	Precision	Recall	F1-score	Support
Optimal	0.95	1.00	0.97	18
Critical	1.00	0.94	0.97	17
Accuracy	-	-	0.97	35
Macro average	0.97	0.97	0.97	35
Weighted average	0.97	0.97	0.97	35

The labels were derived from thresholds applied to the same variables supplied to the classifier. Consequently, the deterministic rule-only baseline reproduced all 35 held-out labels by construction, whereas the Random Forest reproduced 34 (Table 8). This comparison does not support an independent accuracy advantage for machine learning. Instead, the classifier is retained as a supplementary learned component within the prototype, while the deterministic layer remains authoritative for safety-critical status decisions. The practical contribution therefore lies in the audited integration of pH estimation, model interpretation, dashboard display, anomaly handling, and alert delivery rather than in replacing the explicit aquaculture thresholds.

Table 8. The Deterministic Safety Layer Eliminated The Classifier's Single False-Negative Decision

Decision layer	Accuracy	Critical recall	Critical FNR	Interpretation
Random Forest Classifier	97.14%	94.12%	5.88%	One Critical sample was labelled Optimal
Deterministic rule-only baseline	100%*	100%*	0%*	Exact reproduction of threshold-derived labels

*The rule-only baseline values are exact by construction because the reference labels were generated using the same thresholds. These values do not represent independent predictive performance or verified dashboard-override performance.

Deployment, Interpretation, And Limitations

The main system-level finding is that prediction accuracy alone was insufficient for this dataset. Site 5 showed widespread pH sensor failure, and the classifier missed one Critical test sample. The dashboard therefore combines machine-learning prediction with pH-source selection and a rule-based override. A doubtful sensor pH is replaced by the AI-estimated value when it falls outside the valid physical range or differs from the estimate by more than 1.0 pH unit. The override then evaluates the selected pH and the remaining input parameters against its implemented thresholds. These functions address sensor anomalies and specified threshold breaches, although their operational effectiveness requires further validation.

Figure 4 shows the Streamlit dashboard developed from the trained machine-learning pipeline. The interface presents current sensor readings, AI-estimated pH, pH source selection, water-quality classification, feature-importance output, and status notification in a single view. This design improves practical interpretability because users can monitor pond status through an operational interface rather than interpreting raw prediction outputs from a Python notebook.

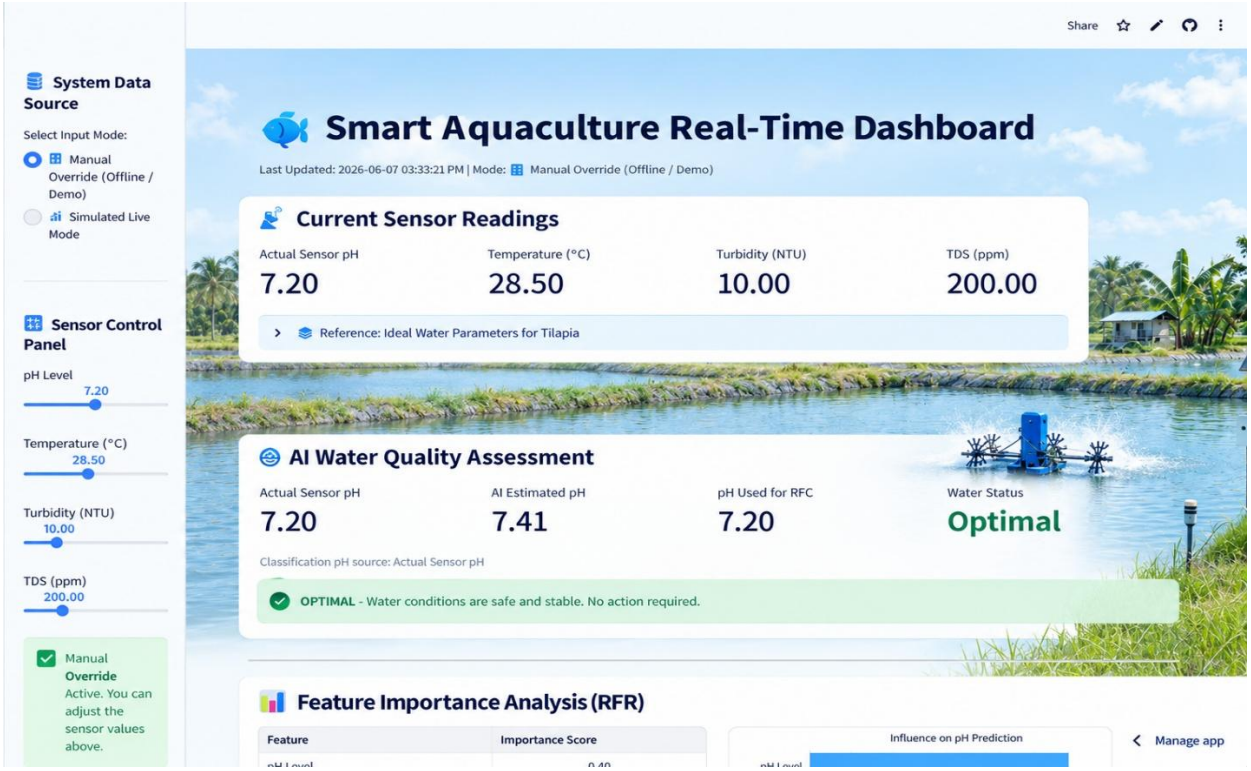


Figure 4. Streamlit dashboard showing demonstration inputs, AI-estimated pH, the pH selected for classification, and the final water-quality status.

The dashboard demonstrates the use of the proposed pipeline as a decision-support prototype for smallholder aquaculture. AI-estimated pH provides a secondary reference for identifying potential sensor anomalies, while

the rule-based override flags threshold breaches in the inputs evaluated by the dashboard. The classifier's single Critical false negative highlights the need to assess safety-related errors alongside overall accuracy. However, the deterministic baseline results do not establish the effectiveness of the implemented dashboard override.

The Telegram alert extends the dashboard from passive visualisation to active notification. When the system identifies Critical water status, the alert message provides diagnostic information, including actual pH, estimated pH, pH used for classification, pH source, temperature, turbidity, TDS, and the reason for the warning. This feature is useful for aquaculture operators who may not continuously monitor the dashboard.

The modelling results are consistent with earlier water-quality studies in which Random Forest handled non-linear relationships in tabular environmental data [3], [4], [12]. Machine-learning approaches have also been used for river and fishpond water-quality classification [6], [7]. The present study differs mainly in workflow: the audit, regression model, feature-importance output, classifier, dashboard, and alert logic are placed in one pipeline. The model itself remains a standard baseline. The practical contribution lies in the quality-control and safety layers around the model, especially after the Site 5 pH failure.

Several limitations define how the reported metrics should be interpreted. The regression set contained 126 samples, the classification set contained 174 samples, and both models were evaluated using a single 80:20 split. Repeated cross-validation was not part of the original experimental protocol; therefore, no unverified mean or standard-deviation estimates are reported. The single-split results may be optimistic and should be treated as exploratory. Training data came only from Sites 1 and 2, leaving cross-site and seasonal transfer untested. Only four parameters were available, while dissolved oxygen, ammonia, nitrite, nitrate, salinity, and conductivity were absent. The records also lacked complete metadata on sampling interval, sensor model, calibration routine, and maintenance history, all of which affect reliability and generalisation.

Future evaluation should rerun the analysis on the row-level dataset using repeated k-fold validation and report mean and standard deviation for R-squared, RMSE, MAE, accuracy, Critical recall, and F1-score. Random Forest should then be compared under identical folds with Decision Tree, support vector machine, XGBoost, and the deterministic baseline. A larger multi-site dataset should cover different seasons, feeding periods, and water-exchange conditions. The dashboard should also be tested through a live sensor endpoint or hardware-in-the-loop setup so that latency, missing packets, sensor drift, and alert timing can be measured under realistic operating conditions.

For small farms, the results support a cautious deployment strategy. A standard tree-based model can provide useful estimates when the input data are screened and every critical decision retains a rule-based fallback. The dashboard can help an operator prioritise TDS inspection, recognise a doubtful pH probe, and receive a Telegram warning when water conditions breach fixed limits. Therefore, the current system should be described as a decision-support prototype rather than a fully automated real-time control system, as it uses manual and simulated inputs and does not control pumps, aerators, or treatment devices.

CONCLUSIONS

This study developed a data-audited machine-learning pipeline for pH estimation and water-status classification in smallholder aquaculture. The audit showed that sensor reliability was not a minor preprocessing issue. Three of five monitoring sites could not support model training: Sites 3 and 4 had too few usable records, while Site 5 showed widespread pH failure. After this screening step, Sites 1 and 2 provided 126 samples for pH regression and 174 samples for water-status classification. The main lesson is that machine-learning performance in IoT aquaculture depends first on whether the sensor data are physically credible.

The Random Forest models met the predefined targets within the cleaned dataset. The regressor estimated pH with R-squared = 0.9337, RMSE = 0.0859, and MAE = 0.0471. TDS carried the largest MDI importance share at 64.50%, followed by temperature and turbidity. The classifier achieved 97.14% accuracy, 94.12% Critical-condition recall, and a 5.88% Critical false-negative rate. However, the deterministic baseline reproduced the

threshold-derived labels exactly by construction. The classifier therefore provides no demonstrated standalone accuracy advantage over the explicit rules and should remain supplementary to the deterministic safety layer.

The meaning of the study is therefore practical rather than purely algorithmic. A monitoring dashboard for small farms should not pass sensor readings directly into a model and treat the output as final. It should first reject physically impossible readings, detect likely pH sensor faults, preserve the same scaling used during training, and override the classifier when hard safety limits are breached. The Streamlit dashboard and Telegram alert demonstrate this workflow at prototype level. The system converts existing sensor records into a clearer operating signal for farm operators while keeping human intervention and threshold rules in the loop.

The findings remain limited to two monitoring sites, 126 regression samples, 174 classification samples, four water-quality parameters, and a single train-test split. The dashboard has not yet been connected to a live sensor endpoint, and the original records do not fully document sampling interval, sensor model, calibration procedure, or maintenance history. Before operational deployment, the pipeline should be revalidated using multi-site seasonal data, repeated cross-validation, independent baseline models, and live or hardware-in-the-loop tests. Within these limits, the study contributes a defensible prototype workflow for turning imperfect aquaculture sensor data into screened, interpretable, and action-oriented decision support.

ACKNOWLEDGMENTS

The authors thank the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM), for supporting this work, and the tilapia-farm operators for providing access to the monitoring records.

DATA AVAILABILITY STATEMENT

The data supporting this study are available from the corresponding author upon reasonable request, subject to permission from the data owner.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

REFERENCES

1. Prapti, D. R., Mohamed Shariff, A. R., Che Man, H., Mohamed Ramli, N., Perumal, T., & Shariff, M. (2022). Internet of Things (IoT)-based aquaculture: An overview of IoT application on water quality monitoring. Reviews in Aquaculture, 14(2), 979–992. https://doi.org/10.1111/raq.12637
2. Rahu, M. A., Chandio, A. F., Aurangzeb, K., Karim, S., Alhussein, M., & Anwar, M. S. (2023). Toward design of Internet of Things and machine learning-enabled frameworks for analysis and prediction of water quality. IEEE Access, 11, 101055–101086. https://doi.org/10.1109/ACCESS.2023.3315649
3. Najah Ahmed, A., Binti Othman, F., Abdulmohsin Afan, H., Khaleel Ibrahim, R., Ming Fai, C., Shabbir Hossain, M., Ehteram, M., & Elshafie, A. (2019). Machine learning methods for better water quality prediction. Journal of Hydrology, 578, Article 124084. https://doi.org/10.1016/j.jhydrol.2019.124084
4. Bui, D. T., Khosravi, K., Tiefenbacher, J., Nguyen, H., & Kazakis, N. (2020). Improving prediction of water quality indices using novel hybrid machine-learning algorithms. Science of the Total Environment, 721, Article 137612. https://doi.org/10.1016/j.scitotenv.2020.137612
5. Lap, B. Q., Phan, T.-T.-H., Nguyen, H.-D., Quang, L. X., Hang, P. T., Phi, N. Q., Hoang, V. T., Linh, P. G., & Hang, B. T. T. (2023). Predicting water quality index (WQI) by feature selection and machine learning: A case study of An Kim Hai irrigation system. Ecological Informatics, 74, Article 101991. https://doi.org/10.1016/j.ecoinf.2023.101991
6. Abdul Malek, N. H., Wan Yaacob, W. F., Md Nasir, S. A., & Shaadan, N. (2022). Prediction of water quality classification of the Kelantan River Basin, Malaysia, using machine learning techniques. Water, 14(7), Article 1067. https://doi.org/10.3390/w14071067

7. Danuri, D., & Pozi, M. S. M. (2024). Machine learning approaches for fish pond water quality classification: Random forest, Gaussian Naive Bayes, and decision tree comparison. In Proceedings of the 11th International Applied Business and Engineering Conference (ABEC 2023). https://doi.org/10.4108/eai.21-9-2023.2342964
8. Alnemari, A. M., Elmessery, W. M., Qazaq, A. S., Moustapha, M. E., Rakhimgaliyeva, S., Abuhussein, M. F. A., Alhag, S. K., Al-Shuraym, L. A., Moghanm, F. S., Szűcs, P., Eid, M. H., & Elwakeel, A. E. (2025). Developing highly accurate machine learning models for optimizing water quality management decisions in tilapia aquaculture. Scientific Reports, 15, Article 35600. https://doi.org/10.1038/s41598-025-16939-w
9. Essamlali, I., Nhaila, H., & El Khaili, M. (2024). Advances in machine learning and IoT for water quality monitoring: A comprehensive review. Heliyon, 10(6), Article e27920. https://doi.org/10.1016/j.heliyon.2024.e27920
10. Lokman, A., Wan Ismail, W. Z., & Ab Aziz, N. A. (2025). A review of water quality forecasting and classification using machine learning models and statistical analysis. Water, 17(15), Article 2243. https://doi.org/10.3390/w17152243
11. Dritsas, E., & Trigka, M. (2023). Efficient data-driven machine learning models for water quality prediction. Computation, 11(2), Article 16. https://doi.org/10.3390/computation11020016
12. Xu, J., Xu, Z., Kuang, J., Lin, C., Xiao, L., Huang, X., & Zhang, Y. (2021). An alternative to laboratory testing: Random forest-based water quality prediction framework for inland and nearshore water bodies. Water, 13(22), Article 3262. https://doi.org/10.3390/w13223262
13. Firdiani, F., Mandala, S., Adiwijaya, A., & Abdullah, A. H. (2024). WaQuPs: A ROS-integrated ensemble learning model for precise water quality prediction. Applied Sciences, 14(1), Article 262. https://doi.org/10.3390/app14010262
14. Hussein, E. E., Jat Baloch, M. Y., Nigar, A., Abualkhair, H. F., Aldawood, F. K., & Tageldin, E. (2023). Machine learning algorithms for predicting the water quality index. Water, 15(20), Article 3540. https://doi.org/10.3390/w15203540
15. Cai, H., Zhang, C., Xu, J., Wang, F., Xiao, L., Huang, S., & Zhang, Y. (2023). Water quality prediction based on the KF-LSTM encoder-decoder network: A case study with missing data collection. Water, 15(14), Article 2542. https://doi.org/10.3390/w15142542
16. Madni, H. A., Umer, M., Ishaq, A., Abuzinadah, N., Saidani, O., Alsubai, S., Hamdi, M., & Ashraf, I. (2023). Water-quality prediction based on H2O AutoML and explainable AI techniques. Water, 15(3), Article 475. https://doi.org/10.3390/w15030475
17. Grbčić, L., Družeta, S., Mauša, G., Lipić, T., Vukić Lušić, D., Alvir, M., Lučin, I., Sikirica, A., Davidović, D., Travaš, V., Kalafatović, D., Pikelj, K., Fajković, H., Holjević, T., & Kranjčević, L. (2022). Coastal water quality prediction based on machine learning with feature interpretation and spatio-temporal analysis. Environmental Modelling & Software, 155, Article 105458. https://doi.org/10.1016/j.envsoft.2022.105458
18. Wang, S., Peng, H., & Liang, S. (2022). Prediction of estuarine water quality using an interpretable machine learning approach. Journal of Hydrology, 605, Article 127320. https://doi.org/10.1016/j.jhydrol.2021.127320
19. Miller, T., Durlik, I., Krzemińska, A., Kisiel, A., Cembrowska-Lech, D., Spychalski, I., & Tuński, T. (2023). Predictive modeling of urban lake water quality using machine learning: A 20-year study. Applied Sciences, 13(20), Article 11217. https://doi.org/10.3390/app132011217
20. Hoque, J. M. Z., Ab Aziz, N. A., Alelyani, S., Mohana, M., & Hosain, M. (2022). Improving water quality index prediction using regression learning models. International Journal of Environmental Research and Public Health, 19(20), Article 13702. https://doi.org/10.3390/ijerph192013702
21. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
22. Naim, S. M., Das, P., Tiang, J.-J., & Nahid, A.-A. (2025). Aquaculture water quality classification using XGBoost classifier model optimized by the honey badger algorithm with SHAP and DiCE-based explanations. Water, 17(20), Article 2993. https://doi.org/10.3390/w17202993
23. Mutiara, G. A., Alfarisi, M. R., & Meisaroh, L. (2026). Smart multisensor-based machine learning for multiclass freshwater fishpond water quality degradation classification. Engineering, Technology & Applied Science Research, 16(3), 34795–34806. https://doi.org/10.48084/etasr.17840

24. Hridoy, M. A. A. M., Shawkat, A. I., Bordin, C., Acharjee, M. R., Masood, A., Baki, A. O., & Al Mamun, M. A. (2025). Advanced machine learning models for accurate water quality classification and WQI prediction: Implications for aquatic disease risk management. Science of the Total Environment, 1008, Article 180965. https://doi.org/10.1016/j.scitotenv.2025.180965
25. Islam, M. M., Kashem, M. A., Alyami, S. A., & Moni, M. A. (2023). Monitoring water quality metrics of ponds with IoT sensors and machine learning to predict fish species survival. Microprocessors and Microsystems, 102, Article 104930. https://doi.org/10.1016/j.micpro.2023.104930