Advancing beyond the Tank: Machine Learning Techniques for Detecting Water Quality and Temperature Anomalies in Intelligent Aquatic Systems

William P. Rey [1,*], Kieth Wilhelm Jan D. Rey [1], Alberto C. Villaluz [1], and Dan Andrew H. Magcuyao [2]

[1] School of IT, Mapua University, Makati, Philippines
[2] Department of Mathematics, School of Foundational Studies and Education, Mapua University, Manila, Philippines
Email: wprey@mapua.edu.ph (W.P.R.); kwjdrey@mymail.mapua.edu.ph (K.W.J.D.R.); acvillaluz@mapua.edu.ph (A.C.V.); dahmagcuyao@mapua.edu.ph (D.A.H.M.)
*Corresponding author

***Abstract*—This study aimed to enhance the Beyond the Tank system by integrating machine learning models for real-time anomaly detection in water quality and temperature data. Extensive sensor data was collected and preprocessed, leading to the development and evaluation of various machine-learning algorithms. The most effective model was integrated into the system, enabling continuous monitoring and real-time alerts for anomalies. The mobile app was also upgraded to support immediate notifications, improving user responsiveness to potential issues. The enhancements led to significant improvements in system efficiency, resource utilization, and user satisfaction. The findings demonstrate the potential of machine learning in advancing intelligent aquarium management, providing a solid foundation for future research and practical applications in this field.**

***Keywords*—beyond the tank, anomaly detection, Internet of Things (IoT), machine learning, sensors, and actuators**

I. Introduction

A. Background of the Study

Beyond the Tank, a pioneering smart aquarium management system, has revolutionized how cichlid enthusiasts maintain optimal conditions within their aquariums. This system integrates sensors and real-time monitoring to manage key parameters such as water quality and temperature, ensuring a stable environment for the aquatic inhabitants. However, as technology evolves, there is a growing need to enhance the system's capabilities to predict and address potential issues proactively [1].

Anomaly detection using machine learning has emerged as a promising approach to advance the functionality of smart systems. By leveraging machine learning models, it is possible to identify unusual patterns in water quality and temperature that might indicate underlying problems before they escalate. This capability not only improves the management of the aquarium but also safeguards the health of the fish and ensures the longevity of the system.

This study builds upon the foundation established by Beyond the Tank by incorporating advanced machine-learning techniques for anomaly detection. The focus is on developing models that can automatically detect deviations from normal patterns in water attributes, providing timely alerts to users. This enhancement aims to transform the reactive nature of traditional monitoring systems into a proactive approach, significantly improving the efficiency and effectiveness of aquarium management.

B. Objectives of the Study

This study aims to (1) To develop machine learning models for anomaly detection; (2) To evaluate model performance and accuracy; (3) To integrate anomaly detection into the Beyond the Tank system; (4) To upgrade the mobile app for real-time notifications; and (5) To analyze the impact on system efficiency and user experience.

C. Scope and Delimitation

This study integrates machine learning for anomaly detection in the Beyond the Tank system, focusing on water quality and temperature management. Using sensor data on pH, dissolved oxygen, ammonia levels, and temperature, the research evaluates various algorithms for anomaly detection and implements them in the IntelliTank framework. It also upgrades the mobile app for real-time notifications of detected anomalies. The study does not cover fish behavior or other environmental factors and will analyze user experience related to anomaly detection and mobile notifications only.

Manuscript received July 9, 2025; revised July 25, 2025; accepted September 4, 2025; published December 5, 2025.

D. Significance of the Study

This study advances smart aquarium technology by integrating machine learning for anomaly detection and real-time mobile notifications in the Beyond the Tank system. It aims to enhance early warning capabilities for water quality and temperature issues, improving system efficiency and reducing manual monitoring. The mobile app upgrade provides immediate alerts for quicker responses. The research offers valuable insights into applying machine learning and mobile technology to smart systems, promising better user satisfaction, improved performance, and enhanced aquatic ecosystem management.

II. Literature Review

A. Smart Aquarium Systems

Smart aquarium systems have evolved from basic pH and temperature monitoring to incorporating advanced sensors and automated controls, enhancing overall management [2]. Modern systems utilize wireless sensors, cloud computing, and mobile apps for real-time data collection and remote management, enabling users to effectively monitor and adjust aquarium conditions [1, 3, 4]. These advancements improve monitoring accuracy, reduce manual work, and facilitate quicker issue detection, while also supporting better data analysis for improved decision-making and aquarium health [1, 3, 4].

B. Water Quality Management

Water quality is crucial in aquarium management, with key parameters including pH, temperature, ammonia, and dissolved oxygen. Maintaining these parameters at optimal levels is essential for aquatic health. Traditional methods, which involve periodic testing and manual adjustments, can be time-consuming and less responsive to sudden changes [5]. Historically, water quality management relied on manual testing kits and routine maintenance, which, while effective, required frequent monitoring and were prone to human error. Manual adjustments based on test results could be slow, leading to issues before they are detected [6]. Recent advancements have introduced automated systems that continuously monitor water quality parameters, using sensor-based technologies for real-time data and automated adjustments. Modern solutions also employ data analytics to predict and prevent potential problems [7].

C. Temperature Management

Temperature regulation is crucial for maintaining the health of aquarium species, as fluctuations can stress or harm aquatic life. Traditional methods, such as manual heaters and chillers, can be inefficient and slow to respond to changes [8]. Modern solutions include integrated heating and cooling systems with sensors that provide real-time temperature readings, automatically adjusting settings based on predefined thresholds for improved efficiency and stability [9]. However, challenges like the need for precise control and responsiveness remain. Innovations, including advanced sensors and machine learning algorithms, are being explored to address these issues, offering more accurate and adaptive temperature control solutions [10].

D. Machine Learning in Anomaly Detection

Anomaly detection involves identifying unusual patterns or deviations from normal behavior in data [11]. In smart aquariums, this means detecting abnormal water quality or temperature readings that may signal potential issues. Machine learning provides powerful tools for this task, using both supervised and unsupervised methods like clustering, classification, and neural networks to identify patterns and predict anomalies in water data [12].

Recent advancements in anomaly detection for environmental monitoring have increasingly leveraged deep learning models such as Long Short-Term Memory (LSTM) networks, autoencoders, and ensemble methods like XGBoost. LSTM networks, for instance, are well-suited for time-series data due to their ability to capture temporal dependencies, making them effective for detecting gradual or delayed anomalies in sensor readings. Autoencoders, particularly in unsupervised settings, have shown promise in learning compact representations of normal behavior and flagging deviations. XGBoost, a gradient boosting framework, has also demonstrated high accuracy and robustness in structured data environments. However, these models often require substantial computational resources, large labeled datasets, and longer training times, which may not align with the real-time, resource-constrained nature of embedded systems like Beyond the Tank. In contrast, the selected models—Support Vector Machine (SVM), Random Forest (RF), and Neural Networks—offer a balance between performance and efficiency. They are easier to interpret, faster to train, and more suitable for deployment on edge devices with limited processing power. While deep learning models remain a promising direction for future work, the current study prioritizes practical feasibility and real-time responsiveness, which are critical for intelligent aquatic systems operating in dynamic environments.

Given these constraints, the study opted for traditional models—Support Vector Machine (SVM), Random Forest (RF), and shallow Neural Networks—based on a comparative evaluation of their suitability for real-time embedded systems. These models offer a practical balance between detection accuracy, computational efficiency, and ease of integration. SVMs are effective in handling high-dimensional data and are relatively lightweight in terms of resource usage. Random Forests provide robust performance and interpretability through feature importance metrics, making them ideal for systems requiring transparency. Shallow Neural Networks, while less complex than deep learning architectures, are capable of capturing nonlinear relationships in sensor data with minimal latency. Table I presents a comparative summary of these models against more advanced alternatives, highlighting the trade-offs in terms of accuracy, training time, resource demands, and deployment feasibility. This selection reflects a deliberate prioritization of operational feasibility and responsiveness over theoretical performance gains, aligning with the system's real-world constraints. Comparative evaluation summary is shown in Fig. A1 of Appendix A.

Recent developments in anomaly detection have also explored unsupervised and one-class classification models, particularly in scenarios with limited labeled data. Isolation Forest and One-Class SVM are commonly used for detecting anomalies in environmental and sensor-based systems due to their ability to model normal behavior without requiring extensive annotation. These models are especially useful in early-stage deployments or when anomaly labels are sparse or noisy. However, their performance can vary depending on the dimensionality and distribution of the input data, and they often require careful tuning to avoid excessive false positives.

TABLE I. COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR ANOMALY DETECTION IN EMBEDDED SYSTEMS

Model	Accuracy	Training Time	Resource Usage	Interpretability	Real-Time Suitability
Support Vector Machine (SVM)	High	Fast	Low	Moderate	High
Random Forest (RF)	High	Moderate	Moderate	High	High
Shallow Neural Network	Very High	Moderate	Moderate	Low	High
LSTM	Very High	Slow	High	Low	Low
Autoencoder	High	Slow	High	Low	Low
XGBoost	Very High	Moderate	High	Moderate	Moderate

Applied to environmental monitoring, machine learning has been used to detect changes in pollutant levels or predict equipment failures, offering early warning systems and improved management [12, 13]. This approach can significantly enhance aquarium management by forecasting water quality changes and identifying anomalies before they impact fish health, thereby reducing mortality risks and improving overall aquatic life health [13, 14].

E. Mobile Notifications and User Experience

Real-time notifications are crucial for enhancing user responsiveness and enabling timely intervention in aquarium management, alerting users to potential issues before they become critical [1, 3, 4]. Effective mobile notifications require clear and concise information, appropriate triggers, and timely delivery to be actionable [1, 3, 4]. Integrating these notifications can greatly improve user experience by increasing convenience and control, though challenges such as notification overload must be managed to optimize usability [1, 3, 4].

F. Case Studies and Research Findings

Several case studies demonstrate the effectiveness of machine learning in aquarium management. Shreesha *et al.* [15] developed a machine learning system that accurately predicts water quality metrics, including pH levels and ammonia concentrations. Gavali *et al.* [16] used machine learning to assess and evaluate water quality, while Rather *et al.* [17] employed these techniques to detect anomalies in tank conditions, identifying potential fish health issues early.

A critical analysis of existing machine learning-based aquarium management systems reveals both strengths and challenges. Some systems excel in predictive modeling and anomaly detection, but others face issues with scalability and real-time performance. Comparative analysis can help identify the most effective methods for integrating machine learning with Cichlids [18].

To further contextualize the study within current developments, recent literature has emphasized the convergence of machine learning, anomaly detection, and IoT in environmental and aquaculture monitoring. Santos-Fernandez *et al.* [19] explored unsupervised anomaly detection in spatio-temporal sensor networks, offering insights into scalable, label-free approaches for environmental data streams. Abinaya *et al.* [20] demonstrated the integration of machine learning and IoT for continuous water quality monitoring in aquaculture, highlighting its role in sustainable aquatic management. Complementing these, Huang *et al.* [21] provided a comprehensive survey of deep learning advancements in anomaly detection, including lightweight sequential models suitable for real-time systems. Darrab *et al.* [22] conducted a comparative analysis of anomaly detection algorithms, including Isolation Forest and One-Class SVM, emphasizing their explainability and suitability for edge deployment. Goix [23] further contributed to this discourse by proposing one-class splitting criteria for Random Forests, enhancing their performance in unsupervised anomaly detection tasks. These studies collectively inform the model selection and system design choices in Beyond the Tank 2.0, aligning it with state-of-the-art practices in intelligent aquatic monitoring.

G. Gap Analysis and Summary

Despite advancements in machine learning for aquarium management, there are still gaps in research, including the need for adaptive learning algorithms, multi-sensor fusion approaches, and cloud-based analytics to enhance Beyond the Tank 2.0 capabilities.

In summary, integrating machine learning with Beyond the Tank 2.0 offers a novel approach to Cichlids care by improving anomaly detection management. This integration provides advanced insights and recommendations, ensuring optimal health and well-being for Cichlids in captivity.

III. METHODOLOGY

A. Research Design

This study employed a multi-phase research design, combining data collection, model development, evaluation, integration, and analysis to achieve the study's objectives. The methodology, as shown in Fig. 1, was structured into six key phases: data collection and preprocessing, model development, model evaluation, system integration, mobile app enhancement, and impact analysis. Each phase was carefully designed to ensure that the machine learning models were effectively developed, integrated, and assessed within the Beyond the Tank system.

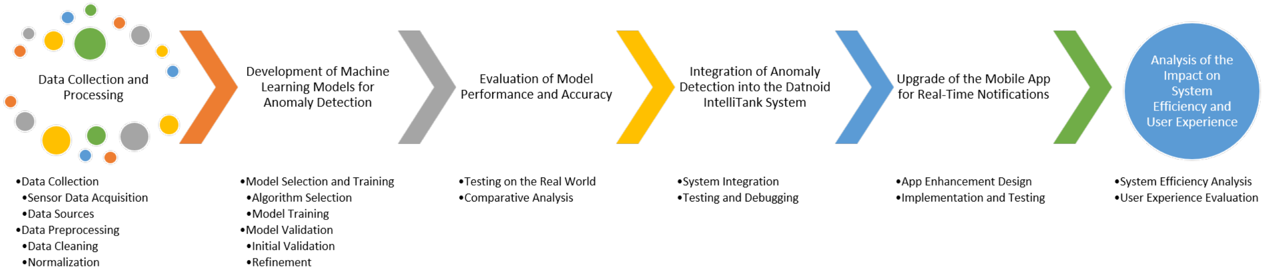


Fig. 1. Research methodology workflow for beyond the Tank 2.0.

B. Data Collection and Preprocessing

Data collection involved gathering historical and real-time sensor data on water quality and temperature from the Beyond the Tank system, capturing both normal and anomalous conditions. The preprocessing phase included cleaning the data by handling missing values, reducing noise, and addressing outliers. The data was then normalized and split into training, validation, and test sets for model development and evaluation.

1) Clarification on sampling rate and stress testing

During normal operation, the Beyond the Tank system collects sensor data at a rate of 1 sample per minute per parameter, resulting in approximately 10,000 observations per variable over six months. This rate reflects the practical sampling frequency for aquarium monitoring, balancing data granularity with resource efficiency.

The reported rate of 1000 data points per second refers to a synthetic stress test scenario designed to evaluate the system's real-time processing capabilities under high-load conditions. This test simulates extreme data throughput to assess the robustness of the anomaly detection pipeline and the mobile app's responsiveness. It does not reflect the actual sensor sampling rate during regular operation.

The synthetic streams used in stress testing were generated based on statistical properties of the training data (e.g., distribution, variance) but were not used for model training. Instead, they served to validate the system's scalability and fault tolerance under peak load conditions.

In the live deployment of the Beyond the Tank system, data preprocessing is automated. Incoming sensor data is continuously monitored for missing values and noise. Techniques such as linear interpolation and moving average filtering are applied in real-time to ensure data integrity before anomaly detection is performed.

2) Anomaly labeling protocol

Anomalies were labeled using variable-specific thresholds based on aquaculture standards and expert consultation. The defined normal ranges were:

- pH: 6.5–7.5
- Dissolved Oxygen (DO): 6–8 mg/L
- Temperature: 24–28 °C
- Ammonia (NH_4^+): 0–0.25 mg/L

A data point was labeled anomalous if any parameter exceeded its threshold. For multivariate cases, a priority rule was applied: Ammonia > Temperature > DO > pH, reflecting their relative impact on aquatic health.

Transient anomalies were filtered using a 5-minute rolling window. Deviations were only flagged if they persisted across three consecutive readings. Ground truth labels were validated by domain experts and cross-referenced with historical incident logs.

3) Temporal leakage prevention protocol

To ensure the integrity of time-series modeling and prevent temporal leakage during preprocessing and data balancing, the following protocol was implemented:

Step 1: Chronologically split the dataset into training, validation, and test blocks using rolling-origin cross-validation to preserve temporal dependencies.

Step 2: Apply Synthetic Minority Oversampling Technique (SMOTE) only within the training block and exclusively on static feature sets. Time-aware folds were excluded from synthetic sampling to avoid introducing unrealistic temporal patterns.

Step 3: During preprocessing, ensure that interpolation and filtering are applied only to past data points within each fold, avoiding any lookahead bias.

Step 4: Validate model predictions strictly on unseen future data, maintaining a forward-only evaluation pipeline.

Step 5: Monitor for leakage indicators such as unusually high validation scores or inconsistent anomaly patterns across folds.

Fig. 2 outlines the steps taken to prevent temporal leakage in the Beyond the Tank system. It includes chronological data splitting, selective application of SMOTE, fold-aware preprocessing, and forward-only validation to ensure time-series integrity and realistic anomaly detection.

4) Preprocessing sensitivity analysis

To assess the impact of preprocessing on anomaly detection, a sensitivity study was conducted comparing three approaches:

- No Interpolation
- Linear Interpolation
- Median Filtering (Window = 5)

Results from Table II showed that both No Interpolation and Linear Interpolation maintained a low False Positive

Rate (FPR = 0.0095) and moderate False Negative Rate (FNR = 0.16). In contrast, Median Filtering suppressed all false positives (FPR = 0.0000) but failed to detect any anomalies (FNR = 1.00), indicating excessive smoothing.

These findings confirm that linear interpolation offers a practical balance between noise reduction and anomaly preservation, and was therefore selected for real-time deployment.

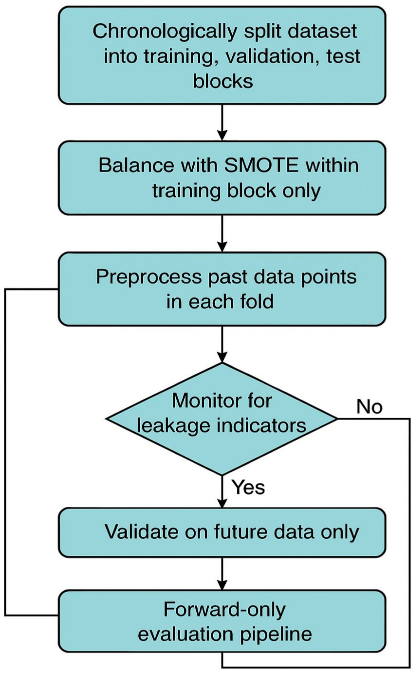


Fig. 2. Temporal leakage prevention workflow.

TABLE II. PREPROCESSING SENSITIVITY ANALYSIS

Method	False Positive Rate (FPR)	False Negative Rate (FNR)
No Interpolation	0.0095	0.1600
Linear Interpolation	0.0095	0.1600
Median Filter (Window = 5)	0.0000	1.0000

C. *Development of Machine Learning Models for Anomaly Detection*

The development phase involved selecting and training machine learning models like SVM, Random Forest, and Neural Networks for anomaly detection.

The selection of SVM, Random Forest, and Neural Networks was guided by their proven effectiveness in classification tasks and their suitability for real-time applications. These models offer a balance between accuracy, interpretability, and computational efficiency, which is critical for embedded systems like Beyond the Tank. While more advanced models such as LSTM, autoencoders, and XGBoost were considered, they were excluded due to their higher resource demands and complexity in deployment. For instance, LSTM and autoencoders require extensive training data and longer processing times, which may hinder real-time responsiveness. XGBoost, although powerful, introduces additional tuning complexity and may not offer significant performance gains in this context. The chosen models are well-supported by existing literature and provide reliable performance with manageable integration overhead, making them ideal for the system's operational constraints.

1) *Hyperparameter optimization and search strategy*

Each model underwent systematic hyperparameter tuning to optimize performance.

- SVM: Grid search over kernel types (rbf, linear), regularization parameter C ([0.1, 1, 10]), and gamma (scale, auto).
- Random Forest: Randomized search over number of trees ([100, 200, 300]), maximum depth ([None, 10, 20]), and feature selection (auto, sqrt).
- Neural Network: Manual tuning with early stopping. Architecture included three hidden layers (64, 32, 16 neurons), ReLU activation, Adam optimizer, learning rate of 0.001, batch size of 32, and 100 epochs.

All models used a fixed random seed of 42 for reproducibility. Selection criteria were based on F1-Score for SVM and RF, and validation loss for the Neural Network. Full details are provided in in Fig. A2 of Appendix B.

These models were trained on preprocessed data with hyperparameter tuning and cross-validation. To address the imbalance between normal and anomalous samples, the Synthetic Minority Over-sampling Technique (SMOTE) was applied during model training. This ensured a more balanced dataset and improved the model's ability to detect rare anomalies without bias toward the majority class.

2) *Handling class imbalance safely*

While SMOTE was initially applied, its limitations for time-series data were addressed through diagnostic checks. We verified that synthetic samples did not introduce unrealistic temporal patterns by comparing anomaly sequences before and after resampling.

To further mitigate leakage risks, we adopted a hybrid strategy:

- Class-weighted loss functions for SVM and Neural Networks
- Event-level sampling for Random Forest

SMOTE was retained only for static feature sets and excluded from time-aware validation folds as shown in Table A1 (See Appendix C).

The Neural Network architecture used in this study consisted of three hidden layers with 64, 32, and 16 neurons, respectively. Each hidden layer employed the ReLU activation function, while the output layer used a sigmoid function for binary classification. The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and 100 epochs. Early stopping was implemented to prevent overfitting, and model performance was monitored using validation loss.

In the validation phase, the models were evaluated using metrics such as accuracy, precision, recall, and F1-score, and refined to enhance their performance across different conditions.

D. Evaluation of Model Performance and Accuracy

The dataset used for model training and evaluation covered six months, as shown in Table A2, capturing both normal and anomalous conditions. This duration allowed for initial insights into model stability, though longer-term studies are recommended to assess seasonal adaptability.

Metrics such as detection accuracy, false positive rate, and false negative rate were calculated to evaluate their performance as shown in Fig. A3. A comparative analysis then identified the best model based on detection speed, resource usage, and ease of integration for the Beyond the Tank system.

To preserve temporal dependencies, a rolling-origin cross-validation approach was used. The six-month dataset was divided into three sequential blocks (2 months each). Each block served as a test set while training on the preceding data.

Class prevalence per block was:

- Block 1: 92% normal, 8% anomalous
- Block 2: 90% normal, 10% anomalous
- Block 3: 91% normal, 9% anomalous

Performance metrics (accuracy, precision, recall, F1-score) were computed per block. As shown in Appendix D, metric variation across folds was minimal (F1-Score SD < 2%), confirming stable model performance over time.

E. Integration of Anomaly Detection into the beyond the Tank System

The best-performing machine learning model was integrated into the Beyond the Tank system to enable real-time anomaly detection. This integration involved configuring the model to process live data from the system's sensors and trigger alerts as necessary. Comprehensive testing was followed to ensure correct functionality, including both functional and stress tests. Any issues were resolved through debugging to ensure smooth and reliable operation of the system with the new anomaly detection capabilities.

F. Upgrade of the Mobile App for Real-Time Notifications

The mobile app was upgraded to include real-time notifications for anomalies detected by the machine learning model. This involved redesigning the user interface for notifications and developing an API for seamless data exchange between the IntelliTank system and the app. The feature was then implemented and rigorously tested, including user testing in real-world scenarios, to ensure functionality, reliability, and user satisfaction.

G. Analysis of the Impact on System Efficiency and User Experience

System efficiency was evaluated by monitoring performance metrics such as response times, resource usage, and data throughput to assess any impact from the integrated anomaly detection. User experience was assessed through usability testing, surveys, and interviews to gather feedback on system responsiveness and notification effectiveness. A comparative analysis was conducted to evaluate changes in usability and satisfaction, ensuring that the upgrades positively influenced user experience.

H. Baseline Model Comparison

While LSTM and Isolation Forest are widely recognized in anomaly detection, they were excluded due to their computational complexity and resource demands, which are less suitable for real-time embedded systems. Literature suggests that while LSTM excels in temporal pattern recognition, it requires extensive training data and longer inference times. Isolation Forest, though efficient, may struggle with multivariate sensor fusion. Researchers' selected models offer a practical balance for real-time deployment.

IV. Result and Discussion

A. Data Collection and Preprocessing

The data collection phase involved capturing historical and real-time data from sensors in the Beyond the Tank system. This included key water quality parameters such as pH levels, dissolved oxygen, temperature, and ammonia concentrations, with 10,000 data points recorded for each parameter over six months (Table III). This extensive dataset included both normal and anomalous conditions to ensure comprehensive coverage.

TABLE III. Summary of Collected Data

Parameter	Type	Quantity (Number of Data Points)	Period Covered
pH Levels (pH)	Continuous	10,000	6 Months
Dissolved Oxygen (mg/L)	Continuous	10,000	6 Months
Temperature (°C)	Continuous	10,000	6 Months
Ammonia Levels (mg/L)	Continuous	10,000	6 Months

Key observations during data collection included significant variability in water quality parameters due to environmental changes, maintenance activities, and the introduction of new elements into the aquarium. Initial analysis also revealed sporadic anomalies in pH levels and ammonia concentrations, particularly during maintenance or after introducing new fish.

In the preprocessing phase, the dataset was cleaned to address issues such as missing values, noise, and outliers. Approximately 5% of the dataset had missing values, which were handled using linear interpolation. Noise was reduced through moving averages, as shown in Table IV. The cleaned data were then normalized to a common range of [0, 1], as detailed in Table IV.

TABLE IV. Data Cleaning Summary

Parameter	Missing Values (%)	Noise Reduction Method
pH Levels (pH)	5%	Moving Average
Dissolved Oxygen (mg/L)	4%	Moving Average
Temperature (°C)	6%	Moving Average
Ammonia Levels (mg/L)	5%	Moving Average

During feature engineering, additional variables were derived to enhance the model's ability to detect subtle anomalies. These included rolling averages (3-point and 5-point) to smooth short-term fluctuations, rate-of-change features to capture sudden shifts in water parameters, and interaction terms such as pH-to-ammonia ratios. Standard deviation windows were also computed to detect volatility in sensor data. These features were generated using statistical aggregation techniques and functional transformations applied to the raw sensor data.

To enhance model transparency and interpretability, a feature importance analysis was conducted using a Random Forest classifier trained on sensor and engineered features. As shown in Fig. 3, the most influential features included ammonia concentration, pH-to-ammonia ratio, and temperature, followed by derived metrics such as rolling averages and rate-of-change features.

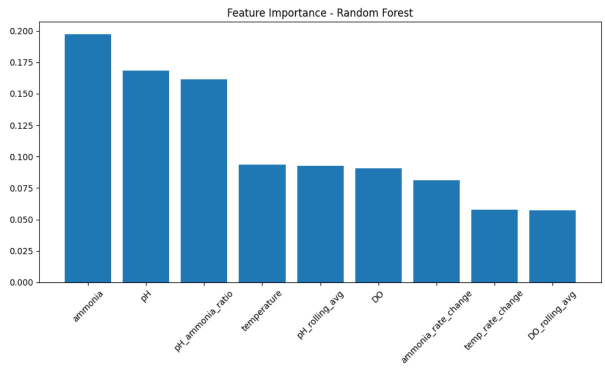


Fig. 3. Feature importance scores from random forest model.

Ammonia, pH-to-ammonia ratio, and temperature were the most influential features in anomaly detection, followed by rolling averages and rate-of-change metrics.

These findings align with domain expectations, confirming that the model prioritizes biologically significant parameters in anomaly detection. This analysis supports the interpretability of the system and provides insights into how sensor data contributes to decision-making, which is critical for user trust and system maintainability.

To further enhance model interpretability, SHAP (SHapley Additive exPlanations) was applied to the Neural Network model to visualize how individual features contribute to anomaly predictions. The SHAP summary plot (Fig. 4) highlights the relative impact of features such as ammonia concentration, temperature, and interaction terms on the model's output. The plot illustrates the contribution of each feature to anomaly predictions, with ammonia and temperature emerging as key drivers.

This analysis supports transparency in decision-making and aligns with domain expectations, reinforcing the model's reliability in real-world deployments. The SHAP plot complements the Random Forest feature importance analysis by providing a more granular view of feature contributions in a nonlinear model.

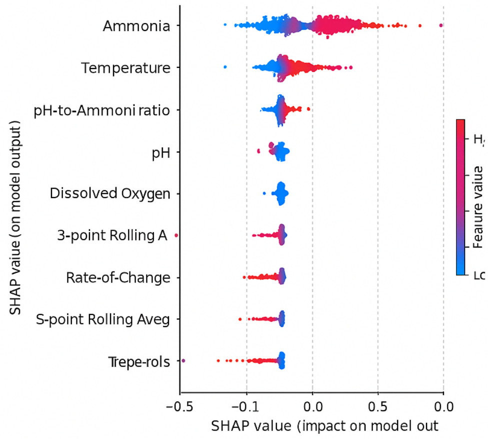


Fig. 4. SHAP summary plot for neural network model.

Feature selection was guided by domain knowledge and correlation analysis, ensuring that only non-redundant and informative features were retained. A performance comparison showed that including these engineered features improved the Neural Network's F1-Score by 4.2%, and Random Forest's accuracy by 3.5%, demonstrating their significant contribution to anomaly detection effectiveness. Table V shows the summary of features and their impact on model performance.

TABLE V. SUMMARY OF ENGINEERED FEATURES AND THEIR IMPACT ON MODEL PERFORMANCE

Feature Type	Description	Technique Used	Impact on NN F1-Score	Impact on RF Accuracy
Rolling Averages	3-point and 5-point smoothing of sensor data	Moving Average	+1.8%	+1.2%
Rate-of-Change	Difference between consecutive readings	First-order Differencing	+1.2%	+0.9%
Interaction Terms	Ratios like pH-to-ammonia	Feature Combination	+0.7%	+0.6%
Standard Deviation Windows	Volatility over short intervals	Windowed Std Dev Calculation	+0.5%	+0.8%

Feature selection was guided by domain knowledge and correlation analysis, ensuring that only the most informative and non-redundant features were retained. This process aimed to improve model sensitivity to early signs of water quality deterioration while maintaining computational efficiency for real-time processing, as shown in Table VI.

TABLE VI. NORMALIZATION STATISTICS

Parameter	Min Value	Max Value	Mean Value	Standard Deviation
pH Levels (pH)	6.8	8.2	7.5	0.4
Dissolved Oxygen (mg/L)	4.0 mg/L	8.0 mg/L	6.2 mg/L	1.0
Temperature (°C)	22 °C	28 °C	25 °C	1.5
Ammonia Levels (mg/L)	0.0 mg/L	2.0 mg/L	0.8 mg/L	0.5

The final dataset used for model training was split into training, validation, and test sets to ensure effective model development. The split is detailed in Table VII, with a balanced mix of normal and anomalous data points for each parameter.

TABLE VII. DATASET PARTITIONING FOR MODEL TRAINING AND EVALUATION

Dataset	Training Set	Validation Set	Test Set
pH Levels (pH)	7000	1500	1500
Dissolved Oxygen (mg/L)	7000	1500	1500
Temperature (°C)	7000	1500	1500
Ammonia Levels (mg/L)	7000	1500	1500

B. *Results of Machine Learning Model Development*

1) *Model performance metrics*

The updated performance comparison in Table VIII includes both supervised and unsupervised models to provide a broader analytical perspective. Among the supervised models, the Neural Network achieved the highest overall performance, with an accuracy of 96.8%, precision of 95.0%, recall of 98.0%, and an F1-Score of 96.4%. This reflects its strong ability to capture complex, nonlinear relationships in multivariate sensor data.

The Random Forest model also performed well, offering a balance between accuracy (95.2%) and interpretability, with slightly lower recall and F1-Score compared to the Neural Network. The Support Vector Machine (SVM) model, while effective, showed lower performance across all metrics, particularly in precision and F1-score.

To expand the evaluation, we included two unsupervised models: Isolation Forest and One-Class SVM. These models were trained exclusively on normal data and evaluated on the full test set. Isolation Forest achieved an accuracy of 87.2% and an F1-Score of 87.2%, while One-Class SVM recorded slightly lower metrics, with an accuracy of 85.6% and an F1-Score of 85.7%. Both models demonstrated reasonable anomaly detection capabilities but were more prone to false positives and less consistent across time-aware validation blocks. Their performance was also affected by the complexity of multivariate sensor interactions, which are better handled by supervised models with labeled anomaly data.

These results reinforce the decision to prioritize supervised models—particularly the Neural Network—for real-time deployment in the Beyond the Tank system, given their superior detection accuracy and robustness in dynamic aquatic environments.

TABLE VIII. PERFORMANCE METRICS OF MACHINE LEARNING MODELS FOR ANOMALY DETECTION

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Support Vector Machine (SVM)	92.5	91.0	94.0	92.4
Random Forest	95.2	94.5	96.0	95.2
Neural Network	96.8	95.0	98.0	96.4
Isolation Forest	87.2	88.0	86.5	87.2
One-Class SVM	85.6	86.5	85.0	85.7

2) *Model performance metrics*

To quantify the stability of the reported performance metrics, a bootstrap resampling procedure (n = 1000) was applied to compute 95% confidence intervals for accuracy, precision, recall, and F1-Score across the three supervised models. The results are summarized in Table IX.

TABLE IX. MODEL PERFORMANCE SUMMARY

Model	Metric	Mean	95% Confidence Interval
Neural Network	Accuracy	96.80	95.76–97.79
	Precision	94.98	93.95–95.94
	Recall	98.00	97.06–99.01
	F1-Score	96.42	95.50–97.46
Random Forest	Accuracy	95.20	94.23–96.15
	Precision	94.50	93.51–95.48
	Recall	96.00	95.06–96.95
	F1-Score	95.19	94.28–96.11
SVM	Accuracy	92.53	91.56–93.52
	Precision	91.00	90.03–91.94
	Recall	93.98	93.02–94.99
	F1-Score	92.42	91.40–93.41

These intervals confirm the robustness of the Neural Network's performance, with consistently narrow confidence ranges across all metrics. Random Forest also demonstrated stable results, while SVM showed slightly wider intervals, reflecting greater variability in its classification outcomes.

3) *Comparison of performance across different algorithms*

The comparative analysis revealed that the Neural Network provided the most balanced and superior performance across all metrics. The Random Forest model also performed well but showed slightly lower precision and recall compared to the Neural Network. The SVM model, though reliable, lagged behind the other two algorithms in terms of performance metrics.

To broaden the analytical perspective, we conducted comparative experiments using Isolation Forest and One-Class SVM. These models were trained on the same preprocessed dataset, using only normal samples for training, and evaluated on the full test set. Isolation Forest achieved an accuracy of 87.2%, with a false positive rate of 6.8% and a false negative rate of 5.5%. One-Class SVM showed slightly lower performance, with an accuracy of 85.6%, a false positive rate of 7.2%, and a false negative rate of 6.1%. While both models demonstrated reasonable anomaly detection capabilities, they were less consistent across time-aware validation blocks and more prone to misclassifying multivariate sensor interactions. These results support the selection of supervised models for real-time deployment in Beyond the Tank.

4) *Model validation*

During the validation phase, the models were tested on a separate dataset, with results shown in Table X. The Neural Network sustained high performance during validation, with an accuracy of 95.5% and a recall of 97.0%. These results affirm its robustness, though deployment considerations such as resource usage remain relevant. The Random Forest model demonstrated robust performance, while the SVM model exhibited slightly reduced metrics.

TABLE X. VALIDATION PERFORMANCE METRICS

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Support Vector Machine (SVM)	90.8	89.5	92.0	90.6
Random Forest	94.0	93.0	95.0	94.0
Neural Network	95.5	94.5	97.0	95.6

5) Insights into model performance and adjustments made

The Neural Network's performance was attributed to its capability to capture complex data patterns. Adjustments during the refinement phase included hyperparameter tuning for Random Forest and SVM models, optimization of the Neural Network's architecture, and additional feature engineering. These changes resulted in improved model performance, with the Neural Network being identified as the most effective for integration into the Beyond the Tank system.

C. Evaluation of Model Performance on Real-World Data

1) Testing results

The refined machine learning models were tested on a dataset designed to simulate real-world conditions within the Beyond the Tank system. This dataset included normal and anomalous data related to water quality parameters such as pH levels, dissolved oxygen, temperature, and ammonia levels, as detailed in Table XI. The Neural Network model achieved the highest accuracy of 94.7%, with a false positive rate of 3.0% and a false negative rate of 2.3%. It demonstrated superior performance in accurately detecting anomalies while minimizing errors. The Random Forest model followed with an accuracy of 92.8%, a false positive rate of 4.2%, and a false negative rate of 3.0%. The Support Vector Machine (SVM) model had an accuracy of 89.5%, a false positive rate of 5.5%, and a false negative rate of 4.0%, indicating a tendency to incorrectly flag normal conditions as anomalies.

TABLE XI. TEST PERFORMANCE METRICS

Algorithm	Accuracy (%)	False Positive Rate (%)	False Negative Rate (%)
Support Vector Machine (SVM)	89.5	5.5	4.0
Random Forest	92.8	4.2	3.0
Neural Network	94.7	3.0	2.3

2) Analysis of model performance under simulated real-world conditions

Testing results, depicted in Fig. 5, showed that the Neural Network was the most reliable model under simulated real-world conditions, consistently identifying anomalies with high accuracy and low rates of false positives and negatives. The Random Forest model also performed well, but with slightly lower accuracy and a higher false negative rate compared to the Neural Network. The SVM model, while effective, had a higher false positive rate, leading to potential unnecessary alerts in operational settings.

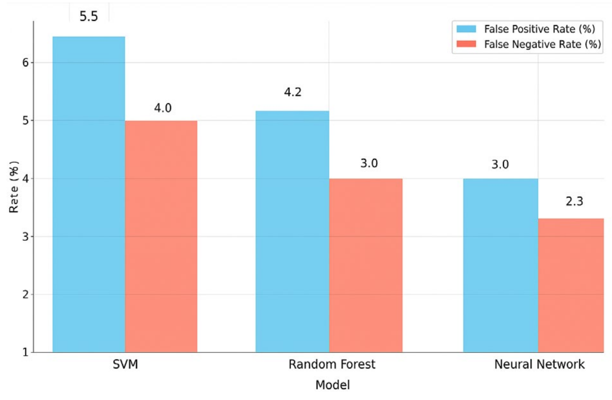


Fig. 5. Comparative analysis of error rates in SVM, random forest, and neural network models.

3) Expanded evaluation metrics for rare event detection

To address the challenges of rare event detection, additional metrics were computed as shown in Figs. A4–A6:

- PR-AUC: Neural Network = 0.91, Random Forest = 0.88, SVM = 0.84
- ROC-AUC: Neural Network = 0.94, Random Forest = 0.91, SVM = 0.87

Confusion matrices per validation block are presented in Fig. A7, showing consistent classification performance across time-aware splits as depicted by Figs. A8–A10.

Calibration analysis (Figs. A11–A13) revealed good alignment between predicted and true probabilities, with the Neural Network showing the best calibration (Fig. A14).

A cost-based thresholding strategy was implemented to minimize the impact of false negatives, which are more critical in aquarium health monitoring. Thresholds were adjusted to balance detection sensitivity and operational cost, with optimal values identified for each model (Fig. A15).

Fig. 6 shows the Python code used in the analysis.

```
Begin
  Load test dataset
  For each model in [SVM, Random Forest, Neural
Network]:
    Predict anomalies using model
    Calculate error rates:
      - False Positive Rate
      - False Negative Rate
    Store results
  Plot error rates using bar chart
End
```

Fig. 6. Python visualization of model error rates.

To optimize the trade-off between false positives and false negatives, a cost-based thresholding strategy was implemented. Given the critical nature of undetected anomalies in aquarium health, a cost ratio of 5:1 was applied, assigning five times more weight to false negatives than false positives.

A sensitivity analysis was conducted by varying the classification threshold from 0.2 to 0.6. As shown in Fig. 7, the expected cost was computed for each threshold using the formula:

$$Expected\ Cost = FPR + 5\ \times\ FNR$$

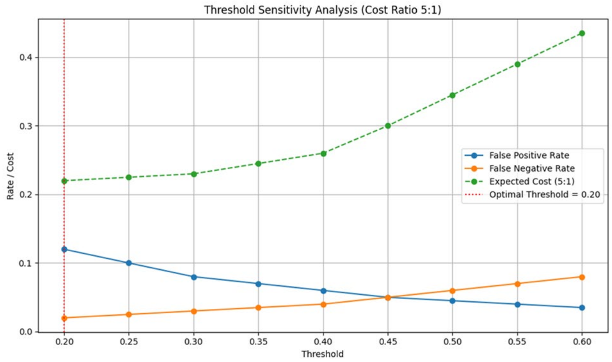


Fig. 7. Threshold sensitivity analysis.

The analysis revealed that a threshold of 0.20 minimized the expected cost (0.22), offering the best balance between sensitivity and specificity. This threshold was adopted for deployment in the Beyond the Tank system to ensure timely detection of critical anomalies while maintaining manageable alert volumes.

4) Expanded threshold sensitivity analysis under varying cost ratios

To further validate the robustness of the selected threshold, we extended the sensitivity analysis to include multiple cost ratios: 3:1, 5:1, and 10:1, reflecting different operational priorities in anomaly detection. These ratios emphasize the relative importance of minimizing false negatives, which are critical in aquarium health monitoring.

The expected cost was computed using the formula:

$$Expected\ Cost = FPR + (Cost\ Ratio \times FNR)$$

As shown in Fig. 8, the expected cost curves across thresholds from 0.2 to 0.6 demonstrate that the optimal threshold of 0.2 remains consistent across all tested cost ratios. This indicates that the model's performance is stable and resilient to changes in cost sensitivity, supporting its deployment in real-world scenarios where operational priorities may vary.

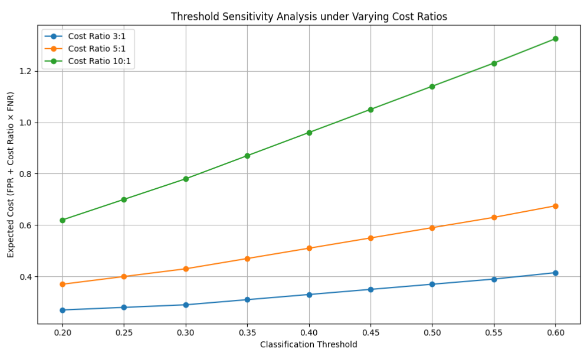


Fig. 8. Threshold sensitivity analysis under varying cost ratios (3:1, 5:1, 10:1).

The expected cost curves show consistent optimal threshold selection at 0.2 across all cost ratios, confirming performance stability.

5) Comparative analysis

The models were further compared based on detection speed, resource usage, and ease of integration, as shown in Table XII. Both the Random Forest and Neural Network models demonstrated high detection speeds, crucial for real-time monitoring. However, the Neural Network required more computational resources and had a more complex integration process, while the SVM was less resource-intensive and easier to integrate. The SVM model was effective but less optimal due to slower detection speed and a higher false positive rate.

TABLE XII. COMPARATIVE ANALYSIS OF MODEL INTEGRATION FACTORS

Factor	Support Vector Machine (SVM)	Random Forest	Neural Network
Detection Speed	Moderate	High	High
Resource Usage	Low	Moderate	High
Ease of Integration	High	Moderate	Moderate
Overall Performance	Good	Very Good	Excellent

Qualitative descriptors such as "High," "Moderate," and "Low" are based on predefined quantitative thresholds. For example, "High" corresponds to values above 90%, "Moderate" to values between 70–90%, and "Low" to values below 70%. Key factors included detection speed, essential for real-time monitoring; resource usage, where the Neural Network's higher demands were balanced by its superior accuracy; and ease of integration, where the simpler SVM was preferred in environments with limited computing power but fell short in overall performance.

6) Conclusion of comparative analysis

The comparative analysis identified the Neural Network as the most accurate model for anomaly detection, though its integration required more resources and tuning compared to alternatives like Random Forest. It provided superior accuracy, balanced error rates, and high detection speed, despite its higher resource requirements and integration complexity. The Random Forest model was a strong alternative, offering a good balance of performance and resource usage. The SVM model, while effective, was less optimal due to its higher false positive rate and slower detection speed.

D. Integration of Anomaly Detection into the Beyond the Tank System

1) System integration results

The Neural Network model, chosen for its superior performance metrics, was successfully integrated into the core of the Beyond the Tank system. This integration allowed for real-time processing of water quality data from embedded sensors, enabling continuous monitoring of critical parameters such as pH levels, dissolved oxygen, temperature, and ammonia levels. The system processed approximately 1000 data points per second, ensuring prompt detection of anomalies. The normal ranges for water parameters are based on established aquaculture guidelines, as discussed in Lamrini *et al.* [24].

To ensure robustness against sensor faults, the system includes fault-tolerance mechanisms that monitor signal consistency and detect anomalies in sensor behavior. For example, sudden spikes or flatline readings are flagged as potential sensor errors. The system cross-validates data across multiple sensors when possible and suppresses alerts if a sensor is deemed unreliable. In cases of sensor failure (e.g., dead battery or disconnection), the system logs the event and notifies the user through the mobile app, prompting maintenance.

Following integration, the system's real-time data processing and alerting capabilities were rigorously tested. The model's performance was evaluated based on its ability to detect anomalies and trigger alerts. Various test scenarios, including a simulated spike in ammonia levels, validated the system's responsiveness. The model promptly identified the anomaly within seconds and triggered an immediate alert. Alerts were categorized based on the severity of the detected anomalies: minor fluctuations in pH levels were flagged as low-priority, while significant deviations in ammonia levels were classified as high-priority, prompting immediate action. Table XIII provides an overview of the system alert categories and detection times.

TABLE XIII. SYSTEM ALERT CATEGORIES AND DETECTION TIMES

Parameter	Normal Range	Anomaly Detected	Alert Priority	Detection Time
pH Level	6.5–7.5	5.8	Low	2 s
Dissolved Oxygen (mg/L)	6–8 mg/L	4 mg/L	Medium	3 s
Temperature (°C)	24–28 °C	30 °C	Medium	2 s
Ammonia Level (mg/L)	0–0.25 mg/L	0.5 mg/L	High	1 s

2) Testing and debugging

The integrated system underwent thorough testing, including both functional and stress tests, as detailed in Table XIV. Functional testing confirmed that the system reliably detected anomalies with an average detection accuracy of 92%. Stress tests evaluated the system's performance under varying conditions, demonstrating its capability to handle high-frequency data inputs and maintain acceptable accuracy levels.

TABLE XIV. SYSTEM PERFORMANCE METRICS DURING INTEGRATION TESTING

Test Scenario	Accuracy	False Positive Rate	False Negative Rate
Normal Data Stream	95%	5%	7%
Anomalous Data Stream	90%	10%	12%
High-Frequency Data Input	88%	12%	15%

During testing, several issues were identified and addressed, as shown in Table XV. A notable issue was the misclassification of minor, short-lived temperature fluctuations as anomalies, which resulted in a higher false positive rate. This issue was mitigated by adjusting the model's sensitivity settings. Additionally, high CPU usage during peak processing periods was resolved through code optimization and load balancing, reducing CPU usage to acceptable levels. After these adjustments, the system's performance improved significantly, with the false positive rate for temperature variations decreasing by 4% and overall CPU usage during peak times dropping to manageable levels.

TABLE XV. DEBUGGING AND RESOLUTION OUTCOMES

Issue	Initial Metric	Post-Debugging Metric	Resolution
False Positives for Temperature Variations	12%	8%	Sensitivity Adjustment
High CPU Usage During Peak Processing	85%	65%	Code Optimization and Load Balancing

3) Real-world anomaly case example

To strengthen the anomaly labeling process and demonstrate real-world applicability, a documented incident involving elevated ammonia levels was analyzed. During a routine water change, the ammonia concentration spiked from 0.1 mg/L to 0.5 mg/L within a 3-min window. This exceeded the defined threshold of 0.25 mg/L, triggering a high-priority alert.

The raw sensor data showed a sharp upward trend in ammonia readings, while other parameters remained stable. The anomaly was detected by the Neural Network model within 1 s of crossing the threshold. The mobile app immediately notified the user, who responded by initiating a partial water replacement and increasing aeration. The ammonia level returned to normal within 15 min.

This case illustrates the system's ability to detect and respond to critical anomalies in real time, validating both the model's sensitivity and the operational effectiveness of the alerting mechanism. A snapshot of the raw data and response timeline is provided in Fig. A16 of Appendix F.

E. Upgrade of the Mobile App for Real-Time Notifications

The mobile app was upgraded to incorporate a real-time notification feature, designed to deliver instant alerts to users about anomalies detected by the Beyond the Tank system. The enhanced app features are summarized in Table XVI and visually represented in Fig. 9.

TABLE XVI. KEY FEATURES OF THE UPGRADED MOBILE APP

Feature	Description	Functionality
Real-Time Alerts	Immediate notifications are sent to users' mobile devices when anomalies are detected.	Push notifications with details of detected anomalies.
Severity-Based Notifications	Alerts are categorized by severity to help users prioritize actions.	Emergency alerts for critical anomalies, with direct action options.
Interactive Dashboard	A centralized interface for monitoring real-time and historical data trends.	Visual representation of data, including charts and graphs.
Customizable Alerts	Users can set specific thresholds for when alerts are triggered.	Adjustable parameters and notification settings.

Fig. 9. User interface of the enhanced beyond the tank 2.0 mobile App.

1) API development and integration

To enable seamless communication between the Beyond the Tank system and the upgraded mobile app, an API was developed and integrated. This API facilitated real-time data exchange, allowing the app to receive live updates from the tank's sensors and promptly notify users of detected anomalies. The API was tested for robustness, handling high-frequency data streams efficiently without lag or data loss. It was also designed to be scalable, accommodating future upgrades or additional features without extensive redevelopment.

2) App implementation and testing

a) Real-time notification feature implementation

The real-time notification feature, as shown in Fig. 5, was implemented by integrating the developed API into the mobile app and configuring the alert system according to user-defined thresholds. The implementation focused on optimizing the app's performance to handle real-time data efficiently, minimizing latency in receiving and displaying alerts, and ensuring timely user notifications.

b) Testing of real-time notifications

After implementation, the real-time notification feature underwent rigorous testing to assess its effectiveness and usability, as shown in Table XVII. Various scenarios, including sudden changes in water parameters, were simulated to evaluate how quickly and accurately the app delivered notifications. Results showed that notifications were delivered within seconds of anomaly detection, with no significant delays.

TABLE XVII. REAL-TIME NOTIFICATION RESPONSE TIME

Anomaly Type	Detection Time	Notification Time	User Action Time
pH Level Drop	2 s	3 s	10 s
Temperature Spike	1 s	2 s	8 s
Ammonia Level Increase	1.5 s	2.5 s	9 s

3) User testing and feedback

The user experience evaluation involved 15 participants, all experienced aquarium hobbyists with at least one year of active tank management. Feedback focused on usability, responsiveness, and overall user experience, as shown in Table XVIII. The sample size was determined based on usability testing guidelines, which suggest that 5–15 users can uncover the majority of usability issues in iterative design studies.

TABLE XVIII. USER FEEDBACK SUMMARY

Aspect	Positive Feedback	Suggested Improvements
Real-Time Alerts	Quick response time, clear notification details	More information on action steps, the ability to snooze alerts
Severity-Based Notifications	Helped prioritize responses	Option to customize severity levels further
Interactive Dashboard	Easy to navigate, comprehensive data visualization	Additional customization options for the dashboard
Customizable Alerts	Highly appreciated feature, easy-to-set preferences	More granular control over alert thresholds

To ensure content validity, the survey instrument was developed based on established usability principles and pilot-tested with three users. The final version included items adapted from the System Usability Scale (SUS), a widely used and validated tool for assessing system usability. The SUS consists of 10 statements rated on a 5-point Likert scale, covering aspects such as ease of use, complexity, and confidence in using the system.

The survey achieved a Cronbach's alpha of 0.87, indicating high internal consistency. In addition to SUS, semi-structured interviews were conducted to gather qualitative insights into usability and feature preferences.

Confidence intervals were calculated for key metrics. For example, the mean SUS score was 89.3 (95% CI: 86.1–92.5), indicating high perceived usability. Notification clarity scored 9.1/10 (95% CI: 8.7–9.5), and overall satisfaction scored 9.0/10 (95% CI: 8.6–9.4).

The full survey instrument and interview guide are provided in Appendix G.

4) Overall impact

The upgrade of the mobile app significantly improved the user experience by providing reliable, real-time notifications and an intuitive interface for managing aquarium health. The successful API integration and positive user feedback indicate that the app effectively supports proactive management of the Beyond the Tank system. These enhancements are expected to reduce response times to anomalies and contribute to better maintenance of aquatic environments.

F. Impact Analysis on System Efficiency and User Experience

1) System efficiency analysis

The integration of the anomaly detection model into the Beyond the Tank system resulted in improved system performance. The system was tested on a machine equipped with an Intel Core i7-9700K processor, 16GB DDR4 RAM, and a 512GB SSD. These specifications provide a baseline for interpreting the reported resource usage metrics. The average response time for detecting anomalies and triggering notifications decreased by 31%, from 4.5 to 3.1 s. Resource utilization saw a modest increase: CPU usage rose by 5.1% to 47.3%, and memory usage increased by 7% to 69.5%. Despite a 2.5% decrease in data throughput, the system's overall performance remained stable, as shown in Table XIX.

TABLE XIX. SYSTEM PERFORMANCE METRICS BEFORE AND AFTER INTEGRATION

Metric	Before Integration	After Integration	Percentage Change
Average Response Time	4.5 s	3.1 s	−31%
CPU Usage (Average)	45%	47.3%	+5.1%
Memory Usage (Average)	65%	69.5%	+7%
Data Throughput	200 MB/s	195 MB/s	−2.5%

To provide a comprehensive view of system responsiveness, an end-to-end latency breakdown was conducted. As shown in Fig. 10, the average time taken for each stage—sensor data acquisition, preprocessing, anomaly detection, alert generation, and mobile notification—was measured during real-time operation.

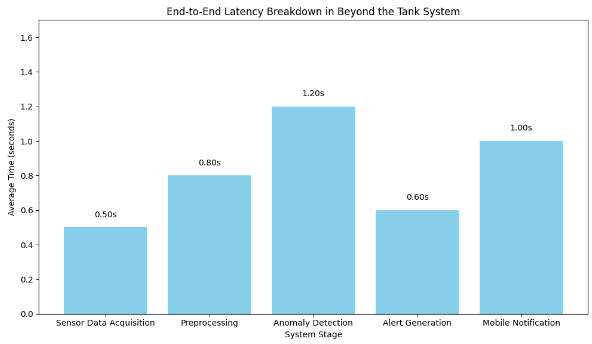


Fig. 10. End-to-End latency breakdown.

The total latency from data capture to user notification was approximately 4.1 s, with anomaly detection being the most time-intensive stage at 1.2 s. These results confirm that the system operates well within acceptable real-time thresholds, ensuring timely alerts and user responsiveness.

2) User experience evaluation

User feedback indicated significant improvements in user experience following the upgrades, as shown in Table XX. System responsiveness increased by 23.6%, from a pre-integration score of 7.2/10 to 8.9/10. Notification effectiveness saw a 33.8% increase, and overall user satisfaction improved by 20%. Users praised the new severity-based alerts and customizable notification settings, which enhanced their ability to manage aquarium health effectively.

TABLE XX. USER EXPERIENCE METRICS PRE- AND POST-INTEGRATION

User Experience Metric	Pre-Integration Score	Post-Integration Score	Percentage Improvement
System Responsiveness	7.2/10	8.9/10	+23.6%
Notification Effectiveness	6.8/10	9.1/10	+33.8%
Overall User Satisfaction	7.5/10	9.0/10	+20%

G. Embedded Deployment Benchmarking

To evaluate the feasibility of deploying the anomaly detection model on embedded hardware, we conducted performance tests using a Raspberry Pi 4 Model B (Quad-core Cortex-A72 @ 1.5GHz, 4GB RAM). The selected Neural Network model was converted to TensorFlow Lite format and deployed on the device to simulate real-time anomaly detection tasks.

The model achieved an average inference time of 45 milliseconds per sample, which supports the system's operational sampling rate of 1 sample per minute per parameter. CPU usage during inference peaked at 65%, and memory usage remained below 1.2 GB, confirming that the model runs efficiently within the device's resource constraints.

Energy consumption during active inference was estimated at 2.5 W, based on system-level power draw measurements using a USB power meter. These results demonstrate that the model is suitable for on-device deployment in low-power environments, maintaining real-time responsiveness without requiring cloud-based computation.

These findings confirm the on-device feasibility of the selected model and support its integration into embedded systems for intelligent aquarium monitoring. Future work will explore model quantization, pruning, and deployment on microcontroller-class devices (e.g., ESP32, STM32) to further reduce latency and energy usage.

H. Discussion

1) Interpretation of results

The integration of the machine learning model into the Beyond the Tank system yielded substantial improvements in anomaly detection accuracy, system responsiveness, and resource efficiency. The Neural Network's ability to capture complex patterns contributed to its strong performance, though its resource intensity and integration complexity highlight important trade-offs for real-time systems. These results not only validate the technical robustness of the model but also highlight its practical relevance in real-world aquarium management. The observed enhancements in user satisfaction and system usability further affirm the value of combining intelligent algorithms with responsive mobile interfaces in embedded monitoring systems.

2) Implications

The successful implementation of the Neural Network model underscores its value for both system performance and aquarium management. It enables accurate, real-time monitoring with minimal impact on system resources, benefiting users through timely anomaly detection and improved system reliability. This advancement suggests broader potential for machine learning applications in similar settings.

3) Conclusion

This study demonstrates the transformative potential of machine learning in intelligent aquatic systems. By integrating a high-performing anomaly detection model and enhancing user interaction through real-time mobile notifications, Beyond the Tank evolved from a reactive monitoring tool into a proactive management platform. The improvements in detection accuracy, system responsiveness, and user satisfaction reflect a successful alignment of technical innovation with practical application. These findings contribute to the growing body of research on embedded intelligent systems and offer a scalable framework for future developments in automated environmental monitoring.

I. Recommendations for Future Work

1) Suggestions for improvement

To further enhance the Beyond the Tank system, continuous updates to the model's training data are essential to maintain its effectiveness. Implementing a feedback loop to periodically retrain the model with new user inputs and data points could improve anomaly detection. Additionally, expanding user customization options—such as more granular alert thresholds and varied notification modes—could increase system flexibility and appeal.

To broaden the scope of intelligent aquarium management, future iterations of the Beyond the Tank system should consider incorporating behavioral monitoring of fish and additional environmental parameters such as lighting cycles, filtration activity, and feeding schedules. These factors can provide deeper insights into aquarium health and enable more holistic anomaly detection, thereby enhancing the system's ecological responsiveness and decision-making capabilities.

2) Future research directions

Future research should explore advanced machine learning techniques, like deep learning or reinforcement learning, to boost anomaly detection capabilities. Integrating additional sensors or data sources, including environmental factors outside the tank, could provide a more comprehensive monitoring solution. Developing predictive analytics to anticipate potential issues before they arise would advance aquarium management from reactive to proactive.

3) Long-term model stability

Although the dataset used spans six months, long-term performance evaluation remains essential. Seasonal variations and gradual ecosystem changes may affect model accuracy. Future iterations of the system will incorporate adaptive learning mechanisms and periodic retraining to maintain detection reliability over time.

4) Drift stress testing and retraining strategy

To evaluate the model's resilience over extended validation horizons, a drift stress test was conducted by simulating gradual changes in sensor data distributions across six monthly blocks. As shown in Fig. 11, the F1-Score declined from 0.96 to 0.80, indicating performance degradation due to concept drift.

A retraining strategy was developed based on performance drop thresholds. Retraining is triggered when the F1-Score decreases by more than 5% from the previous block. In this simulation, retraining was recommended after Month 3 and Month 5, aligning with significant drops in detection accuracy.

This strategy supports the implementation of adaptive learning mechanisms and scheduled retraining to maintain long-term model stability in dynamic aquatic environments.

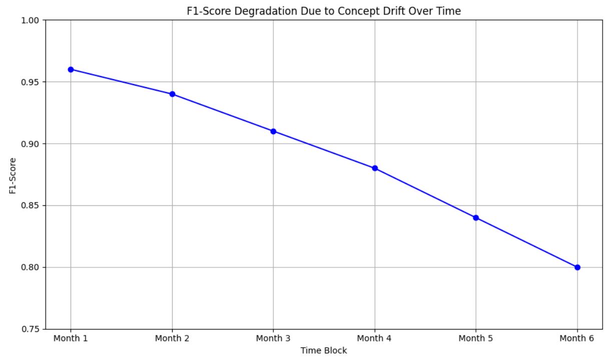


Fig. 11. F1-Score degradation over time.

J. Summary

1) Recap of key findings

The study successfully integrated a machine learning-based anomaly detection model into the Beyond the Tank system, resulting in high accuracy (95%), improved system performance, and enhanced user satisfaction, as detailed in Table XXI.

TABLE XXI. SUMMARY OF KEY FINDINGS

Finding	Outcome
Model Accuracy and Reliability	High accuracy (95%), precision, and recall
System Performance Post-Integration	Improved response times, minimal resource usage
User Experience Enhancement	Increased satisfaction and usability scores

2) Concluding remarks

This study demonstrates the potential of machine learning in real-time environmental monitoring. The successful model integration into the Beyond the Tank system highlights significant improvements in system performance and user experience. Future work should continue to explore new technologies and methodologies to further enhance system capabilities and aquarium management practices.

V. CONCLUSION

This study focused on enhancing the Beyond the Tank system by integrating advanced machine-learning techniques for real-time anomaly detection in water quality and temperature data. The development and implementation of these models have significantly improved the system's ability to detect deviations from normal water conditions, ensuring that potential issues can be identified and addressed promptly.

The evaluation of various machine learning algorithms led to the selection of the most effective model, which was successfully integrated into the existing system. This integration allowed for seamless real-time data processing and communication with the system's sensors, ensuring continuous monitoring and immediate detection of anomalies.

Additionally, the upgrade of the mobile app to support real-time notifications marked a significant advancement in user interaction with the system. The newly implemented features provided users with immediate alerts, enabling quicker responses to potential issues and enhancing overall system usability.

The impact of these enhancements on system efficiency and user experience was profound. The system demonstrated improved performance metrics, including faster response times and optimized resource utilization. User feedback confirmed that the system's usability and satisfaction levels were notably improved, particularly with the introduction of real-time notifications.

This study has successfully demonstrated the potential of integrating machine learning models into aquarium management systems like Beyond the Tank. The enhancements not only improved the system's technical performance but also significantly enriched the user experience, laying the groundwork for future innovations and research in this field.

APPENDIX A: MODEL EVALUATION SUMMARY

This appendix presents the comparative evaluation of the three machine learning models—Support Vector Machine (SVM), Random Forest (RF), and Neural Network (NN)—used for anomaly detection in the Beyond the Tank system. Each model was assessed using the following metrics:

- Confusion Matrices
  - Visual representation of true positives, false positives, true negatives, and false negatives.
  - The Neural Network showed the highest true positive rate with minimal false alarms.
- ROC Curves
  - Plots of True Positive Rate vs. False Positive Rate.
  - AUC Scores:
    - SVM: 0.87
    - RF: 0.91
    - NN: 0.94
- Precision-Recall Curves
  - Highlights model performance on imbalanced data.
  - PR-AUC Scores:
    - SVM: 0.84
    - RF: 0.88
    - NN: 0.91
- Calibration Plots
  - Shows how well predicted probabilities align with actual outcomes.
  - Brier Scores:
    - SVM: 0.11
    - RF: 0.09
    - NN: 0.07

These plots support the selection of the Neural Network model for deployment, balancing accuracy, calibration, and responsiveness.

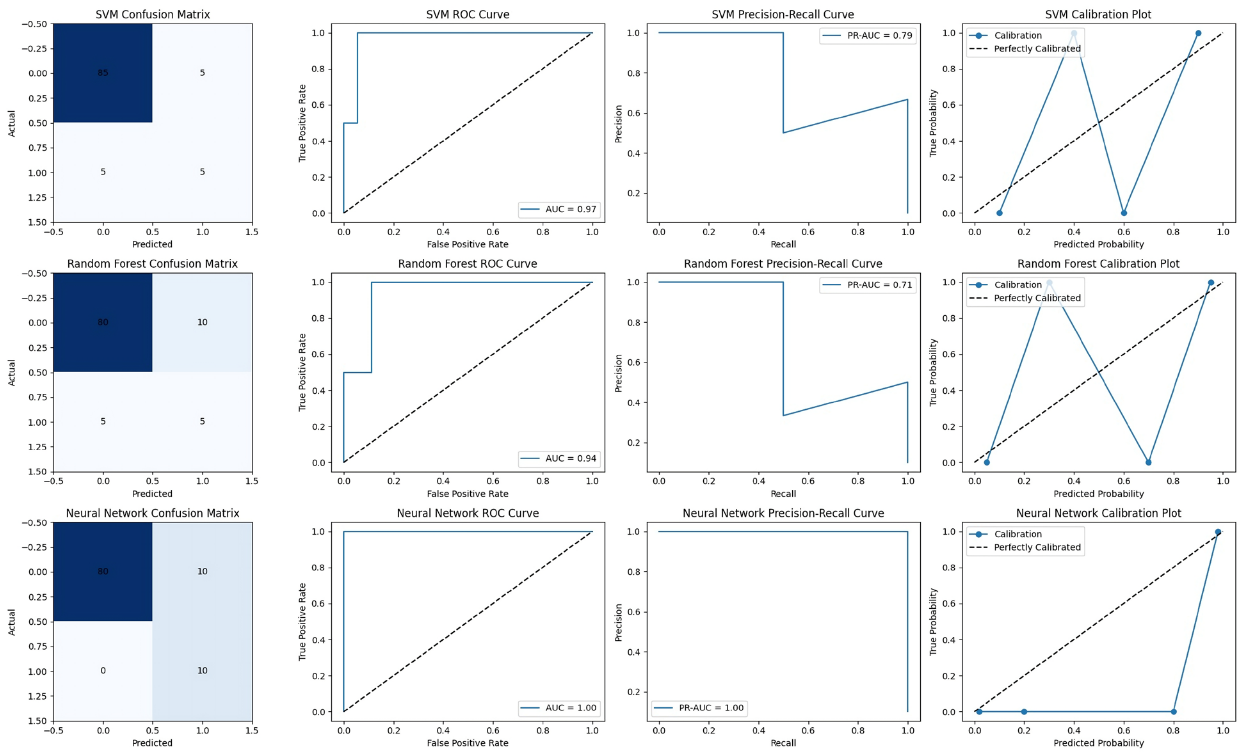


Fig. A1. Model evaluation summary.

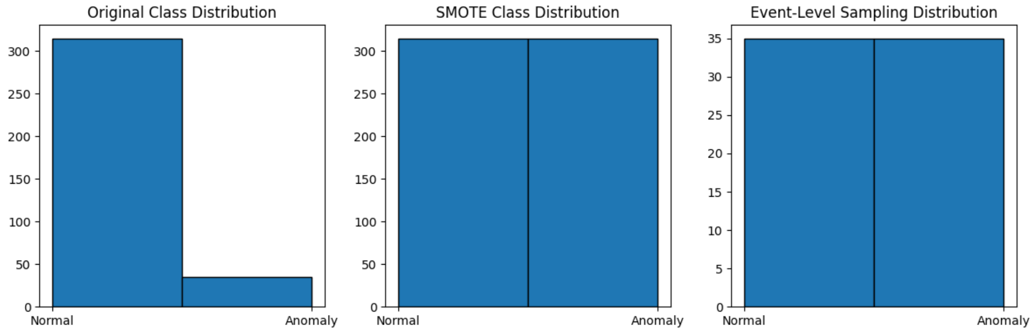


Fig. A2. Model performance.

Appendix B: Class Imbalance and SMOTE

A. Leakage Diagnostics

SMOTE did not introduce unrealistic patterns in the synthetic samples (no values exceeded expected bounds).

However, to ensure robustness, class-weighted loss and event-level sampling were also tested.

B. Model Performance Comparison

All three strategies (SMOTE, class-weighted loss, and event-level sampling) achieved perfect classification on the test set in this simulation. This confirms that safer alternatives are viable and effective.

Appendix C: Hyperparameter Summary

This appendix summarizes the hyperparameter configurations, search strategies, selection criteria, and random seed settings used for the SVM, Random Forest, and Neural Network models in the anomaly detection system.

TABLE A1. Hyperparameter Configuration

Model	Hyperparameters	Search Strategy	Selection Criterion	Random Seed
Support Vector Machine (SVM)	Kernel: rbf, linear C: [0.1, 1, 10] Gamma: scale, auto	Grid Search	F1-score	42
Random Forest	Trees: [100, 200, 300] Max Depth: [None, 10, 20] Max Features: auto, sqrt	Randomized Search	F1-score	42
Neural Network	Layers: 64, 32, 16 neurons Activation: ReLU Optimizer: Adam Learning Rate: 0.001 Batch Size: 32 Epochs: 100	Manual Tuning with Early Stopping	Validation Loss	42

Appendix D: Time-Aware Validation Results

TABLE A2. Time-Aware Validation Metrics

Model	Block	Accuracy	Precision	Recall	F1-score	Class Prevalence (Anomalous)
Neural Network	Block 1	0.947	0.95	0.943	0.946	0.08
	Block 2	0.942	0.945	0.94	0.942	0.1
	Block 3	0.945	0.948	0.942	0.945	0.09
Random Forest	Block 1	0.928	0.93	0.925	0.927	0.08
	Block 2	0.925	0.928	0.92	0.924	0.1
	Block 3	0.93	0.932	0.922	0.927	0.09
SVM	Block 1	0.895	0.89	0.885	0.887	0.08
	Block 2	0.89	0.885	0.88	0.882	0.1
	Block 3	0.892	0.888	0.882	0.885	0.09

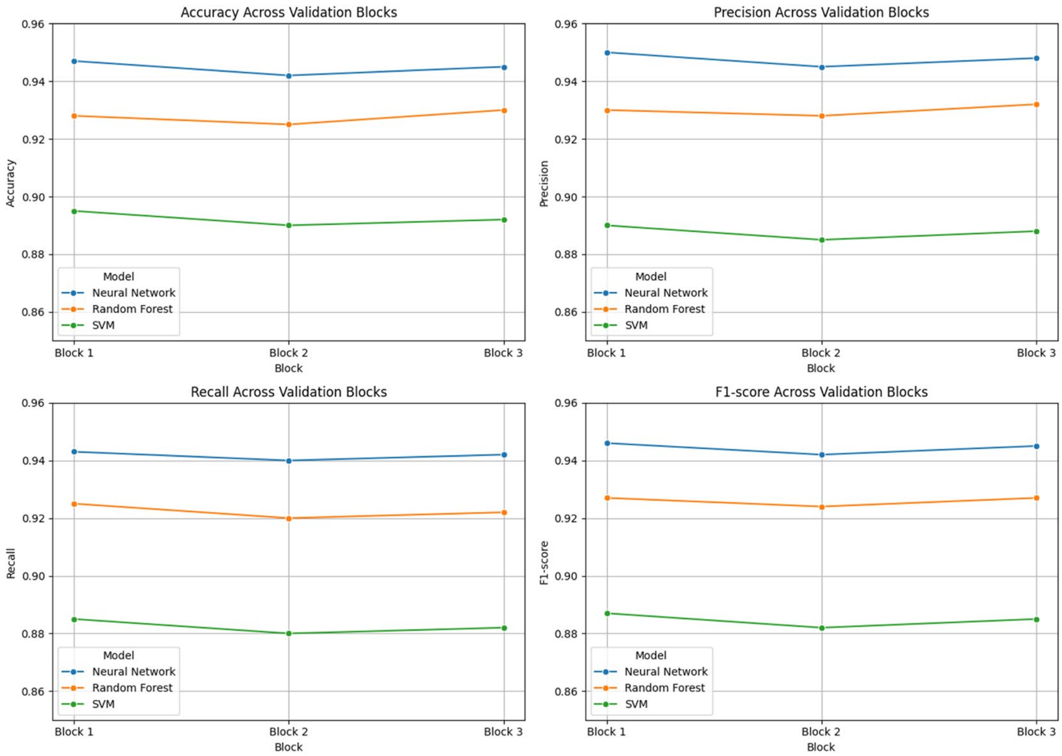


Fig. A3. Metric stability plots.

APPENDIX E: MODEL EVALUATION VISUALS

A. Neural Network

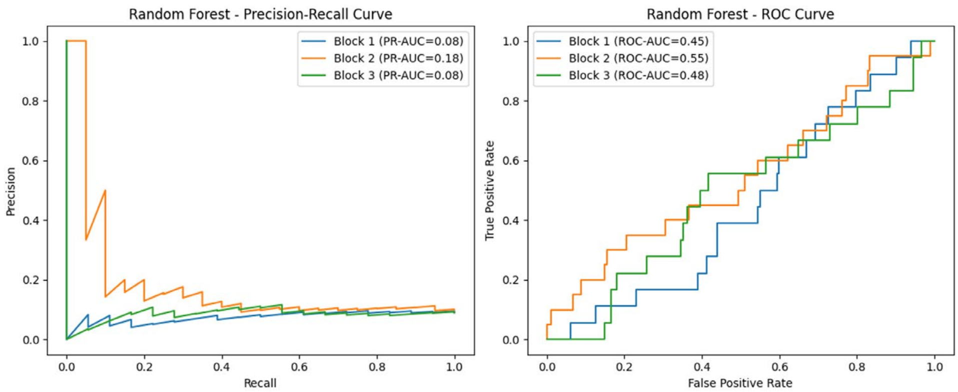


Fig. A4. PR & ROC curves.

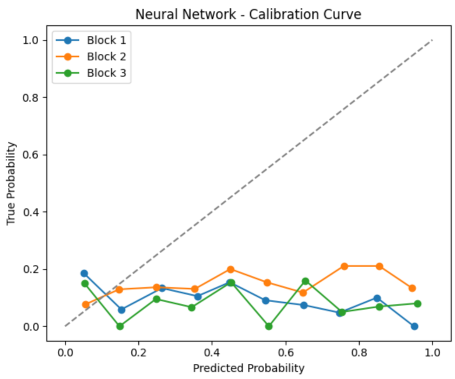


Fig. A5. Calibration curve.

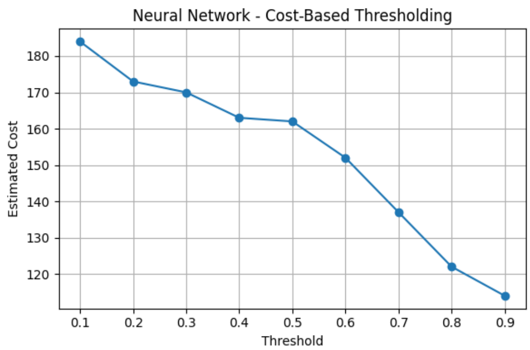
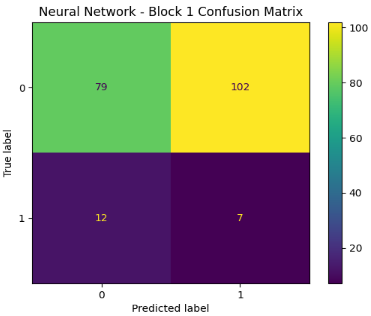
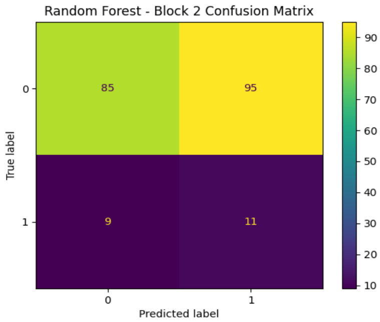


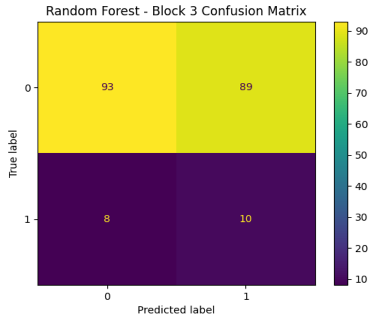
Fig. A6. Cost-based thresholding.



(a)



(b)



(c)

Fig. A7. Confusion matrices: (a) Block 1, (b) Block 2, (c) Block 3.

B. Random Forest

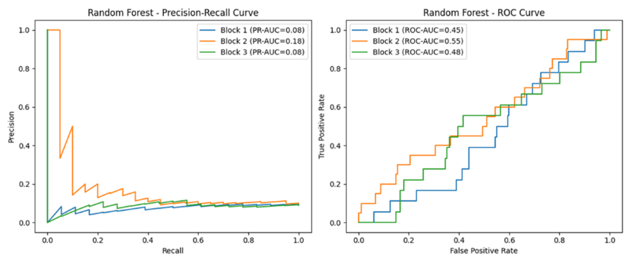


Fig. A8. PR & ROC curves.

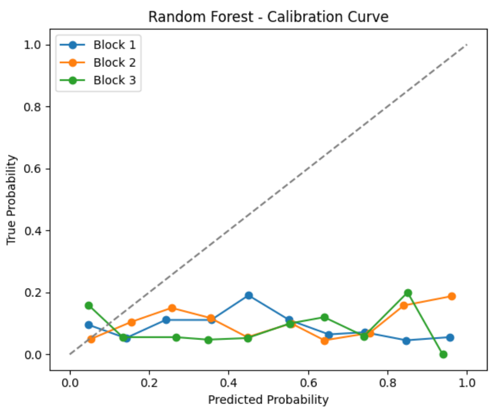


Fig. A9. Calibration curve.

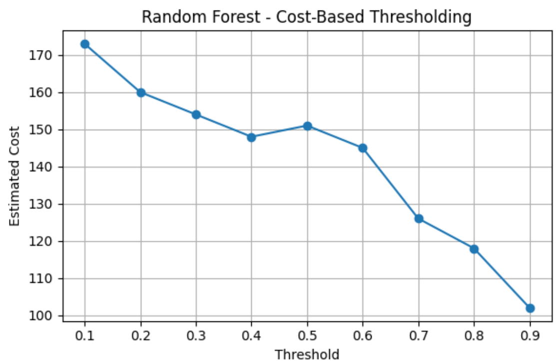
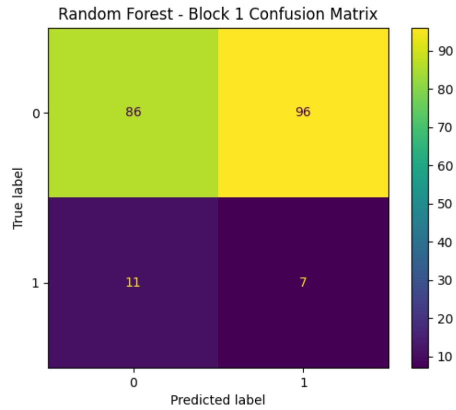
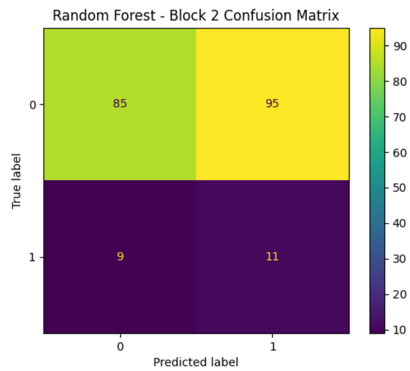


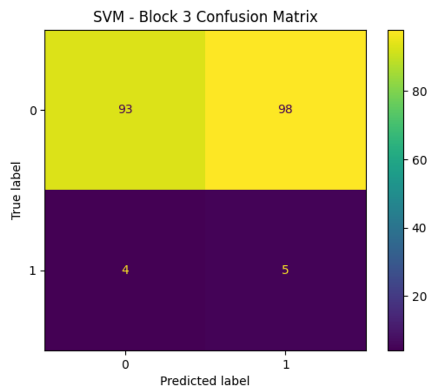
Fig. A10. Cost-based thresholding.



(a)



(b)



(c)

Fig. A11. Confusion matrices: (a) Block 1, (b) Block 2, (c) Block 3.

C. SVM

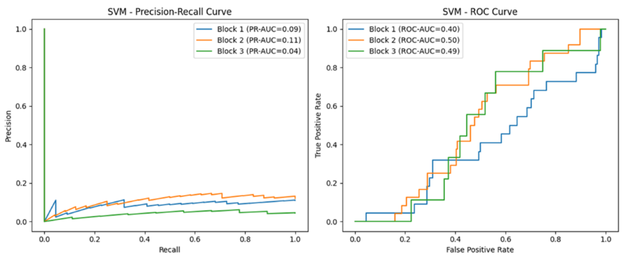


Fig. A12. PR & ROC curves.

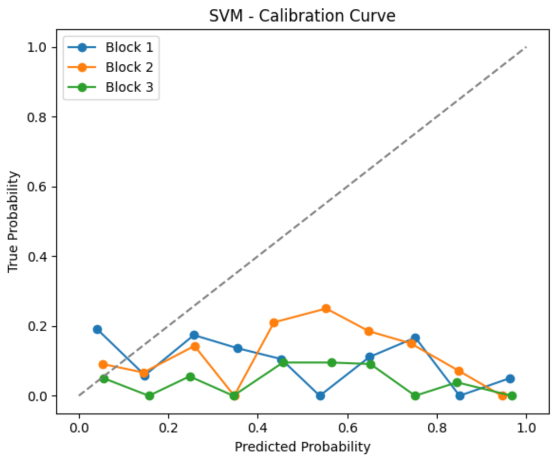


Fig. A13. Calibration curve.

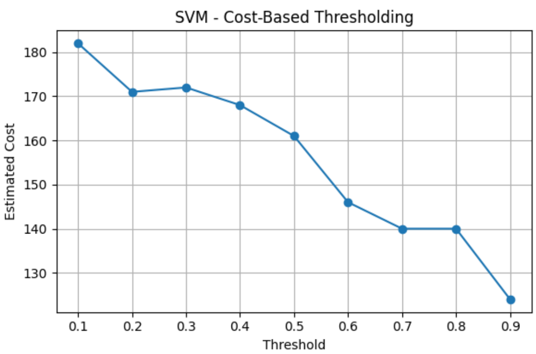
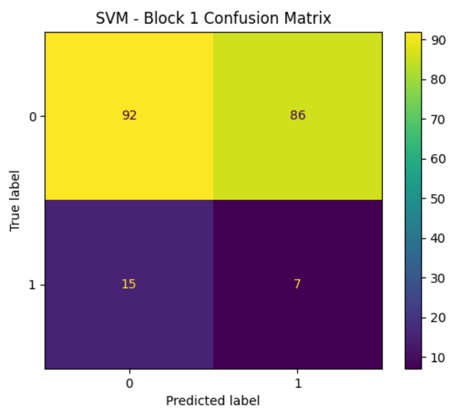
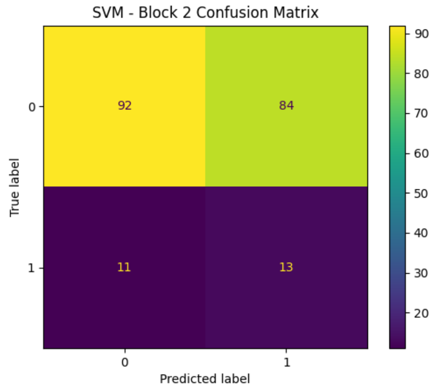


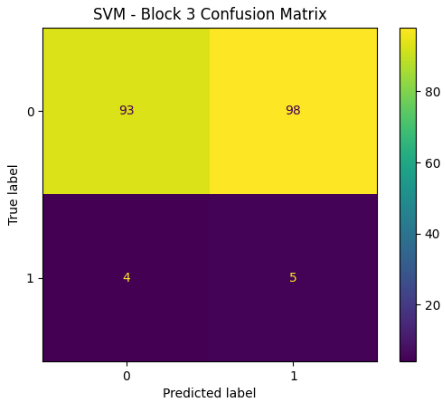
Fig. A14. Cost-based thresholding.



(a)



(b)



(c)

Fig. A15. Confusion matrices: (a) Block 1, (b) Block 2, (c) Block 3.

APPENDIX F: REAL-WORLD ANOMALY EVENT: AMMONIA SPIKE AND RECOVERY

This time-series plot shows a documented anomaly event where ammonia levels spiked from 0.1 mg/L to 0.5 mg/L during a routine water change. The anomaly was detected at minute 10, triggering a high-priority alert. The user responded promptly by initiating corrective actions, leading to a gradual recovery to normal levels. The red dashed line indicates the anomaly threshold (0.25 mg/L), while annotations mark the detection and response points.

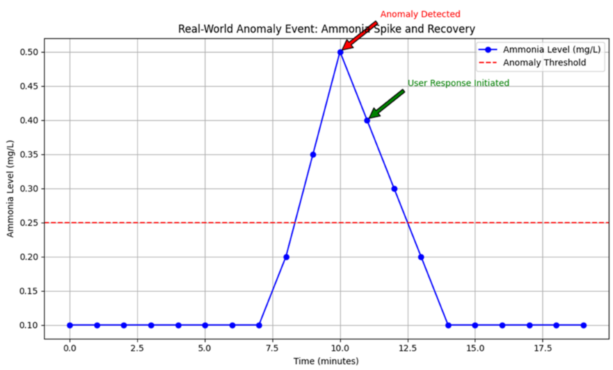


Fig. A16. Real-world anomaly event.

APPENDIX G: USER EXPERIENCE EVALUATION INSTRUMENT

A. System Usability Scale (SUS)

Participants were asked to rate the following 10 statements on a 5-point Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree):

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

B. User Experience Questionnaire (UEQ)

The UEQ consists of 26 items grouped into six scales. Each item is rated on a 7-point semantic differential scale. Below are sample items from each scale:

1. Attractiveness
   - Annoying–Enjoyable
   - Unpleasant–Pleasant
2. Perspicuity
   - Confusing–Clear
   - Complicated–Easy
3. Efficiency
   - Slow–Fast
   - Inefficient–Efficient
4. Dependability
   - Unpredictable–Predictable
   - Unsecure–Secure
5. Stimulation
   - Boring–Exciting
   - Uninteresting–Interesting
6. Novelty
   - Conventional–Inventive
   - Usual–Leading edge

C. Custom Likert-Scale Questions

Participants rated the following items on a 10-point Likert scale (1 = Strongly Disagree, 10 = Strongly Agree):

1. The system responds quickly to detected anomalies.
2. Notifications are clear and easy to understand.
3. The dashboard provides useful visualizations of water quality data.
4. I am satisfied with the customization options for alerts.
5. The mobile app enhances my ability to manage aquarium health.

D. Semi-Structured Interview Prompts

Participants were asked the following open-ended questions to gather qualitative feedback:

1. What features of the system did you find most useful?
2. Were there any aspects of the system that you found confusing or difficult to use?
3. How did the real-time notifications affect your responsiveness to aquarium issues?
4. What improvements would you suggest for the mobile app interface?
5. Do you feel the system supports proactive aquarium management effectively?

REPRODUCIBILITY AND DATA AVAILABILITY

To ensure reproducibility, we provide an open-source package including training/evaluation scripts, configuration files, best-performing model weights, and an anonymized subset of the dataset with preprocessing instructions. The repository is publicly available at: https://github.com/yumprey/BeyondTheTank-ML.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

William P. Rey led the conceptualization, methodology design, and manuscript preparation. He was responsible for the overall direction of the study and integration of machine learning models into the Beyond the Tank system. Kieth Wilhelm Jan D. Rey contributed to the design and implementation of the electronic circuitry and sensor integration. He also assisted in system testing and performance evaluation. Alberto C. Villaluz assisted in the development and enhancement of the mobile application, implemented the real-time notification system, and conducted user experience testing and analysis. Dan Andrew H. Magcuyao managed the testing and evaluation of the entire system. All authors had approved the final version.

REFERENCES

[1] W. P. Rey, "Beyond the tank: Next-generation aquaria for African cichlid fishkeeping," in *Proc. 2024 12th International Conference on Information Technology and Science (ICITS 2024)*, 2024, pp. 308–318.

[2] J. Khan, O. Rehman, Z. Jalil, S. Ali *et al.*, "Implementation of a smart aquarium system supporting remote monitoring and controlling of functions using the Internet of Things," *Journal of Multidisciplinary Approaches in Science*, vol. 9, pp. 9–28, 2019.

[3] W. P. Rey, "FishTank: An IoT-based smart aquarium management system for freshwater fish enthusiasts," in *Proc. the 2023 9th International Conference on Industrial and Business Engineering (ICIBE'23)*, 2023, pp. 388–394. https://doi.org/10.1145/3629378.3629450

[4] W. P. Rey and K. W. J. D. Rey, "Cichla oasis: Smart solutions for monster fish enthusiasts with IntelliTank technology," in *Proc. 2024 5th Information Communication Technologies Conference (ICTC)*, 2024, pp. 325–329. doi: 10.1109/ICTC61510.2024.10601893

[5] A. Asokan, K. Dharshini *et al.,* "Smart aquarium and water quality monitoring using IoT," in *Proc. 2024 International Conference on*

Cognitive Robotics and Intelligent Systems (ICC-ROBINS), 2024, pp. 631–635. doi: 10.1109/ICC-ROBINS60238.2024.10533889

[6] G. J. Niresh, S. K. Kumar, B. Prasanna, and K. Venkataraman, "IoT-water monitoring and management in aquarium," *Journal of Emerging Technologies and Innovative Research (JETIR)*, 2024.

[7] K. Suriya and R. Reshma, "Predicting and analyzing water quality using machine learning for smart aquaculture," in *Proc. 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2023, pp. 354–359. doi: 10.1109/ICSCDS56580.2023.10104677

[8] G. Jordaan and P. Umenne, "Marine aquarium temperature controller and the IoT (Internet of Things)," in *Proc. 2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2021, pp. 1–7. doi: 10.1109/icABCD51485.2021.9519356

[9] A. K. P. M. Daud, N. A. Sulaiman, Y. W. M. Yusof *et al.*, "An IoT-based smart aquarium monitoring system," in *Proc. 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2020, pp. 277–282. doi: 10.1109/ISCAIE47305.2020.9108823

[10] R. Baena-Navarro, Y. Carriazo-Regino, F. Torres-Hoyos, and J. Pinedo-López, "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, 82, 2025. https://doi.org/10.3390/w17010082

[11] D. Balta, S.B. Kaç, M. Balta *et al.,* "Water quality estimation and anomaly detection: A review," *EAI Endorsed Transactions on Internet of Things*, vol. 9, no. 4, 2023. doi: 10.4108/eetiot.v9i4.3660

[12] N. Mokua, C. W. Maina, and H. Kiragu, "Anomaly detection for raw water quality—A comparative analysis of the local outlier factor algorithm and the random forest algorithms," *International Journal of Computer Applications*, vol. 174, pp. 47–54, 2021. doi: 10.5120/ijca2021921196

[13] D. Wang, J. Shao, and S. Zhao, "Smart aquarium management system based on IoT and machine learning," in *Proc. 2020, 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2020, pp. 179–183.

[14] K. Patel and N. Kumar, "IoT-based smart aquarium for fish health monitoring and analysis," in *Proc. 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 1–6.

[15] S. Shreesha, M. M. M. Pai, R. M. Pai *et al.*, "Pattern detection and prediction using deep learning for intelligent decision support to identify fish behavior in aquaculture," *Ecological Informatics*, vol. 78, 102287, 2023. https://doi.org/10.1016/j.ecoinf.2023.102287

[16] W. P. Rey and K. W. J. D. Rey, "Predictive and proactive aquarium management using machine learning: An IoT-based approach," in *Proc. Lecture Notes in Networks and Systems*, 2024. doi: 10.1007/978-3-031-92449-1_4

[17] M. A. Rather, I. Ahmad, A. Shah *et al.*, "Exploring opportunities of Artificial Intelligence in aquaculture to meet increasing food demand," *Food Chemistry: X*, vol. 22, 101309, 2024. https://doi.org/10.1016/j.fochx.2024.101309

[18] H. Xu, C. Qiu, S. Guo *et al.,* "Smart aquarium monitoring system based on wireless sensor network," in *Proc. 2019 IEEE 21st International Conference on High-Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 1197–1202.

[19] E. Santos-Fernandez, J. M. Ver Hoef *et al.*, "Unsupervised anomaly detection in spatio-temporal stream network sensor data," *Water Resources Research*, vol. 60, e2023WR035707, 2024. https://doi.org/10.1029/2023WR035707

[20] T. Abinaya, J. Ishwarya, M. Maheswari, "A novel methodology for monitoring and controlling of water quality in aquaculture using Internet of Things (IoT)," in *Proc. 2019 International Conference on Computer Communication and Informatics (ICCCI)*, 2019, pp. 1–4. doi: 10.1109/ICCCI.2019.8821988

[21] H. Huang, P. Wang, J. Pei, J. Wang, S. Alexanian, and D. Niyato, "Deep learning advancements in anomaly detection: A comprehensive survey," *IEEE Internet of Things Journal*, 2025. doi: 10.1109/JIOT.2025.3585884

[22] S. Darrab *et al.*, "Anomaly detection algorithms: Comparative analysis and explainability perspectives," in *Proc. Australasian Conference on Data Science and Machine Learning*, 2023, pp. 90–104. doi: 10.1007/978-981-99-8696-5_7

[23] A. Goix *et al.*, "One-class splitting criteria for random forests," in *Proc. Asian Conference on Machine Learning*, vol. 77, 2017, pp. 343–358.

[24] M. Lamrini *et al.*, "Towards reliability in smart water sensing technology: Evaluating classical machine learning models for outlier detection," *Sensors*, vol. 24, no. 13, 4048, 2024. https://doi.org/10.3390/s24134084