International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Low-Cost IoT Water Quality Classification Using Three Deployable Sensors and Edge-Deployed Machine Learning

Yash Aggarwal[1], Moksh Sharma[2], Bindu Thakral[3]

[1,2,3]*School of Engineering & Technology, Sushant University, Gurugram, India*

yash.250btccse064@sushantuniversity.edu.in
moksh.250btccse068@sushantuniversity.edu.in
binduthakral@sushantuniversity.edu.in

Abstract - Aquaculture pond health depends on continuous water quality assessment. Dissolved oxygen probes and titration apparatus remain costly for smallholder farmers. Affordable monitoring therefore demands sensors a microcontroller can read directly. Published IoT deployments converge on temperature, pH and turbidity sensing. Most machine learning studies instead train on full laboratory panels. Such accuracy claims assume instrumentation that target farmers rarely own. Our work restricts model inputs to three microcontroller-measurable parameters. Five classifiers and one quantized neural network were benchmarked. A feature-set ablation quantifies accuracy forfeited by excluding laboratory assays. Gradient Boosting reached 87.21% accuracy and 0.8664 macro F1. Restricting inputs to three sensors costs 0.1198 macro F1. Quantized inference occupies 4.37 KB within ESP32 flash limits. Findings give smallholder fish farmers an affordable water monitoring path. Embedded systems researchers can reuse this ablation methodology directly. Reported limitations also guide future field validation on real ponds.

Key Words: Internet of Things, water quality monitoring, aquaculture, edge computing, TinyML, ESP32.

1. INTRODUCTION

Aquaculture supplies a growing share of global protein, and pond water quality directly governs stocking density, feed conversion and mortality. Parameters that matter most to fish health — dissolved oxygen (DO), biochemical oxygen demand (BOD), ammonia, nitrite — are precisely those that are hardest to measure continuously [19]. A laboratory-grade optical DO probe costs orders of magnitude more than a thermistor, and alkalinity and hardness are not sensed at all in the conventional sense: they are determined by acid and ethylenediaminetetraacetic acid (EDTA) titration respectively, which requires reagents, glassware and a trained operator.

Cost asymmetry shapes what is actually deployable. A review of published Internet of Things (IoT) aquaculture deployments shows convergence on a small, consistent instrument set. Islam and Kashem [1] monitored five ponds in Jamalpur District, Bangladesh using sensors and an Arduino microcontroller to collect pH, temperature and turbidity, storing readings on ThingSpeak and analysing them with ten machine learning algorithms across 40,280 rows. Siswanto et al. [2] instrumented an aquaponic system with a pH-4502C probe, a DFROBOT analog total dissolved solids (TDS) sensor and a DS18B20 temperature sensor driven by a NodeMCU ESP8266. Udanor et al. [3] report a comparable sensor complement for catfish aquaponics in Nigeria. Across these independent deployments the same three quantities recur: temperature, pH and turbidity.

Much of the published literature on machine learning for water quality assessment, however, trains on the full laboratory panel and reports accuracy figures that implicitly assume the availability of instrumentation the target user does not own. An accuracy of 98% obtained using ammonia, nitrite, phosphorus and plankton counts describes a laboratory workflow, not a field deployment. A practically useful question is different and is rarely asked directly: how much accuracy is forfeited when the model is restricted to what a microcontroller can read?

This paper answers that question for a three-class aquaculture water quality task. Our contributions are four. First, a quantified cost–accuracy trade-off, where a feature-set ablation isolates the classification penalty attributable to abandoning all wet-chemistry parameters, reported as both an absolute and a relative F1 loss. Second, a deployable-by-construction feature set, in which only temperature, pH and turbidity are used as model inputs while alkalinity and hardness are explicitly classified as laboratory parameters and excluded, a distinction we found is not consistently enforced in comparable work. Third, an edge deployment characterisation, where a trained network is quantized to 8-bit integer (INT8) precision and its memory footprint evaluated against real ESP32 constraints. Fourth, a reported negative result and its diagnosis, since soft-sensing of DO from the deployable inputs fails and we present evidence that the cause lies in the dataset's synthetic construction rather than in sensor inadequacy.

2. RELATED WORK

2.1 IoT Water Quality Monitoring Platforms

Public datasets from deployed IoT aquaculture systems consistently reflect the economics described above. A real-time pond dataset of Islam and Kashem [1] comprises four columns — pH, temperature, turbidity and fish species as the target variable — over 40,280 rows drawn from five ponds. Siswanto et al. [2] published three months of continuous pH, TDS and temperature readings from an ESP8266-based aquaponic installation. Baena-Navarro et al. [7], [8] recorded six months of hourly dissolved oxygen, pH, temperature and turbidity measurements from tilapia ponds in Montería, Colombia, explicitly framing the work as targeting rural environments with limited technological infrastructure, and applied Random Forest and Support Vector Machines for prediction. Recurrence of the same narrow sensor set across four independent deployments on three continents is itself evidence that the constraint we model is real.

2.2 Machine Learning for Water Quality Classification

Tree ensembles dominate reported results on tabular water quality data. Benchmarking evidence supports this: Grinsztajn et al. [6] found that tree-based models remain state-of-the-art on medium-sized tabular datasets of roughly 10,000 samples, even before accounting for their superior training speed, and attributed the gap to differing inductive biases — in particular the robustness of tree ensembles to uninformative features and their ability to learn irregular decision boundaries. Our dataset, at 4,300 samples with 14 heterogeneous numeric features, falls squarely within this regime, and our results reproduce the expected ordering. Classifiers evaluated here follow standard formulations: gradient boosting [10], random forests [11], the scalable boosting variant of [12], and support vector

machines with kernel expansion [13].

2.3 TinyML and Edge Inference

Running inference on the sensing node rather than the cloud reduces bandwidth, removes dependence on continuous connectivity and lowers energy per decision. TensorFlow Lite Micro [5] provides an interpreter-based inference framework designed for embedded resource constraints, with an evaluation demonstrating low resource requirements and minimal runtime overhead. Post-training integer quantization [9] reduces model weights from 32-bit floating point to 8-bit integers, which matters on microcontrollers lacking a floating-point unit. We adopt both, though we are careful in Section 6 to state the limits of what our footprint measurements establish. Standardised benchmarking practice for this class of workload is surveyed in [17], and practical deployment considerations for microcontroller inference are treated in [18].

3. SYSTEM ARCHITECTURE

Fig -1 shows the proposed four-tier architecture. A guiding design principle is that classification occurs at the edge, so that the uplink carries a decision rather than a raw measurement stream.

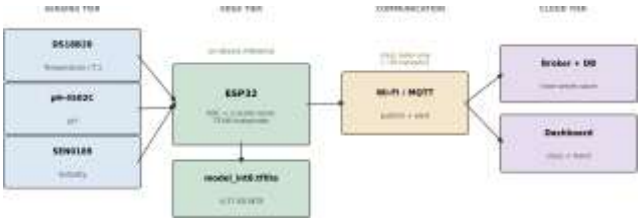


Fig -1: Four-tier architecture. Inference runs on the ESP32; only the resulting class label is transmitted, reducing uplink payload to roughly 20 bytes per transmission and permitting continued operation during connectivity loss.

3.1 Sensing Tier

Three probes are used. A DS18B20 digital sensor provides temperature over a one-wire bus. A pH-4502C module provides an analog pH signal, consistent with the instrumentation reported by Siswanto et al. [2]. A turbidity sensor provides optical clarity. Combined bill of materials is a small fraction of that for a single optical DO probe, and critically requires no reagents or consumables.

3.2 Edge Tier

An ESP32 samples the three channels, applies z-score normalisation using constants exported at training time, and invokes a quantized neural network through the TensorFlow Lite Micro interpreter [5]. Normalisation constants used in our implementation are $\mu = [25.727,\ 7.708,\ 38.817]$ and $\sigma = [9.932,\ 1.599,\ 20.786]$, ordered as temperature, pH, turbidity.

3.3 Communication and Cloud Tiers

A predicted class is published over Message Queuing Telemetry Transport (MQTT). Because the payload is a label rather than a sample vector, transmission cost is largely independent of sampling rate, and a node may buffer decisions locally during network outages. Cloud tier stores the label time series and renders a dashboard; it is not in the inference path.

4. DATASET AND METHODOLOGY

4.1 Dataset

We use the Aquaculture Water Quality Dataset published by Veeramsetty, Arabelli and Bernatin [4]. It contains 1,500 poor, 1,400 excellent and 1,400 good quality samples — 4,300 in total with 14 input features and one output label — and was prepared from threshold values defining acceptable, desirable and stress ranges for fish growth. Classes are encoded 0 (Excellent), 1 (Good) and 2 (Poor). Data contained no missing values and no duplicate rows.

We audited the data for physical plausibility because the reported generation procedure — sampling within per-class threshold ranges — does not guarantee that the joint distribution is physically realisable. Results of that audit, in Table -1, confirm this concern.

Table -1: Dataset physical plausibility audit

Parameter	Range	Min	Max	Violations
Temperature (°C)	[0, 40]	0.19	84.25	100 (2.33%)
pH	[0, 14]	0.00	14.85	16 (0.37%)
DO (mg/L)	[0, 20]	0.13	14.97	0 (0.00%)
BOD (mg/L)	[0, 30]	1.00	14.94	0 (0.00%)

2.33% of temperature values exceed 40 °C, reaching 84.25 °C, and 0.37% of pH values exceed the pH scale maximum. These are not measurement outliers but artifacts of range-based synthetic generation.

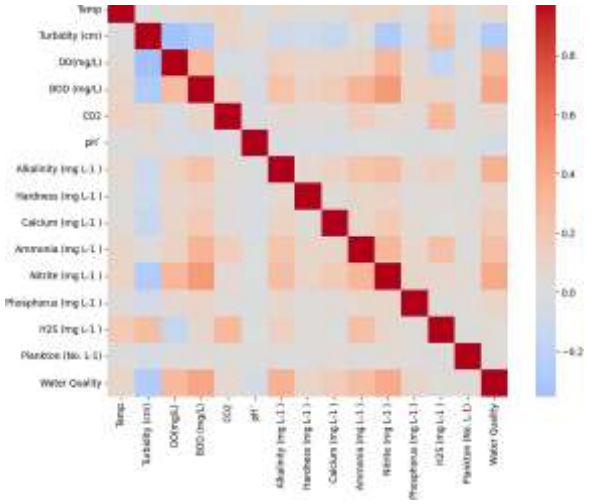


Fig -2: Correlation matrix of all 14 water quality parameters. Near-absence of off-diagonal structure among physically coupled quantities such as temperature and dissolved oxygen is a first indication that the data were generated by independent sampling within class ranges.

A second diagnostic is more consequential. In real water bodies, dissolved oxygen and temperature are strongly negatively correlated through gas solubility. In this dataset the pooled correlation is $r = +0.027$, and within-class correlations are $+0.025$, -0.013 and $+0.031$ for the Excellent, Good and Poor classes respectively. Near-zero within-class structure indicates that features were sampled independently given the label, so essentially all learnable signal resides in the class-conditional marginal ranges. Section 6 discusses the consequences.

4.2 Feature Partition

Table -2 states the partition. A governing criterion is not cost in isolation but measurement modality: whether the quantity can be transduced to an electrical signal by a probe on a microcontroller bus. Alkalinity requires acid titration to a defined endpoint and hardness requires EDTA titration or an ion-selective electrode; neither is a microcontroller sensor, and both are therefore excluded from the model inputs irrespective of their predictive value.

4.3 Methodology Overview

Fig -4 presents the complete processing pipeline as a flowchart. Each stage feeds the next, and a single decision node governs

Table -2: Feature partition by measurement modality

Tier	Parameters	n
Deployable (ESP32)	Temperature, pH, Turbidity	3
Laboratory	DO, BOD, CO_2, Alkalinity, Hardness, Calcium, Ammonia, Nitrite, Phosphorus, H_2S, Plankton	11

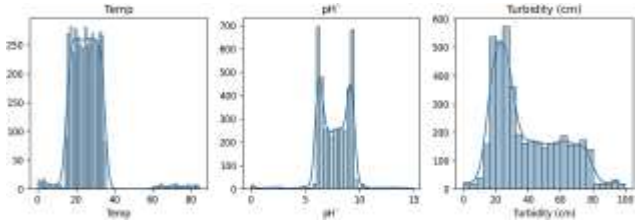


Fig -3: Marginal distributions of the three deployable sensor parameters. Temperature is approximately normal about 25.7 °C but retains a tail beyond 40 °C; turbidity is right-skewed; pH is centred near 7.7. Tails visible here correspond to the plausibility violations in Table -1.

which parameters reach the model.

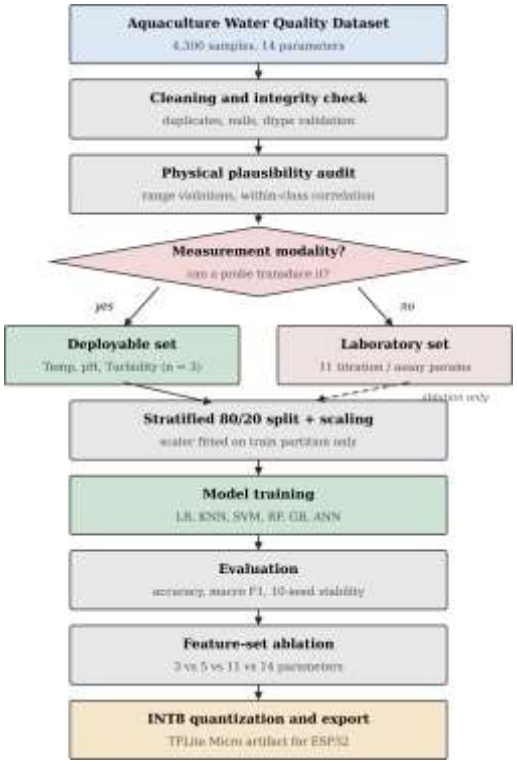


Fig -4: Methodology flowchart. A dashed path indicates that laboratory parameters enter the pipeline only for the ablation study, never for the proposed deployable configuration.

Processing begins with the raw dataset, which passes through cleaning and an integrity check for duplicates, nulls and type consistency. A physical plausibility audit follows, recording range violations and within-class correlations that later justify the negative soft-sensing result. At the decision node each parameter is tested against one question: can a probe on a microcontroller bus transduce it? Parameters answering yes form the deployable set of three; the remaining eleven form the laboratory set. Only the deployable set trains the proposed model, while the laboratory set enters solely through the dashed ablation path. Subsequent stages apply a stratified split, fit the scaler on training data alone, train six model families, and evaluate with accuracy, macro F1 and a ten-seed stability check. A final stage quantizes the trained network to INT8 and exports a TensorFlow Lite Micro artifact sized for ESP32 flash.

4.4 Experimental Protocol

Data were split 80/20 with stratification on the class label and a fixed seed, giving 3,440 training and 860 test samples. Features were standardised using statistics fitted on the training partition only. For cross-validation, scaling was performed inside a pipeline so that fold-specific test data never influenced the scaling statistics — a precaution we note because fitting the scaler on the full array before cross-validation is a common and silent source of optimistic bias.

All models were implemented in scikit-learn [14] and TensorFlow [15], with the neural network optimised using Adam [16] and figures produced with Matplotlib [22]. Five classifiers were evaluated: Logistic Regression, k-Nearest Neighbours ($k = 5$), Support Vector Machine with radial basis function (RBF) kernel, Random Forest (300 trees) and Gradient Boosting. Macro-averaged precision, recall and F1 are reported alongside accuracy, since macro averaging weights the minority behaviour of each class equally. Because a single split yields a point estimate with no error bar, we additionally repeat the classification over ten random splits and report mean and standard deviation. Two regression models were also trained to predict DO and BOD from the same three inputs, evaluated by root mean squared error (RMSE), mean absolute error (MAE) and R^2 with five-fold cross-validation.

5. RESULTS

5.1 Classification Performance

Table -3 reports classification results using only the three deployable sensors. Gradient Boosting is the strongest model at 87.21% accuracy and 0.8664 macro F1. Model ordering — gradient boosting, then random forest, then kernel SVM, then k-NN, with logistic regression far behind — matches the pattern reported by Grinsztajn et al. [6] for medium-sized tabular data. Collapse of logistic regression to 0.5399 F1 indicates that the class boundaries in this three-dimensional feature space are substantially non-linear.

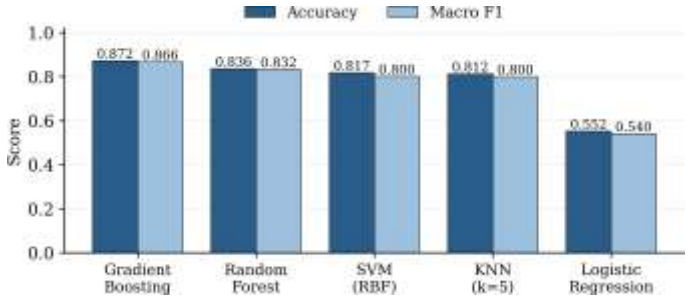


Fig -5: Classification performance on the three deployable sensors. Gradient Boosting leads on both metrics; collapse of logistic regression indicates strongly non-linear class boundaries.

Table -3: Classification performance, three deployable sensors

Model	Acc.	Prec.	Rec.	F1
Gradient Boosting	0.8721	0.8865	0.8775	0.8664
Random Forest	0.8360	0.8368	0.8405	0.8324
SVM (RBF)	0.8174	0.8523	0.8255	0.8003
KNN ($k = 5$)	0.8116	0.8247	0.8185	0.7998
Logistic Regression	0.5523	0.5289	0.5597	0.5399

Precision, recall and F1 are macro-averaged over the three classes.

Table -4 confirms that this ordering is not an artifact of one favourable split. Over ten random splits Gradient Boosting attains 0.8632 ± 0.0061 F1, separated from Random Forest (0.8248 ± 0.0087) by roughly forty standard deviations of run-to-run variation. That comparison is therefore a real difference rather than sampling noise.

Table -4: Stability over ten random splits

Model	Accuracy	Macro F1
Gradient Boosting	0.8688 ± 0.0055	0.8632 ± 0.0061
Random Forest	0.8280 ± 0.0089	0.8248 ± 0.0087
SVM (RBF)	0.8273 ± 0.0068	0.8146 ± 0.0084

Per-class breakdown in Table -5 exposes an operationally important asymmetry. Recall on the Excellent and Good classes is 0.99 and 1.00, but recall on the Poor class is only 0.65 — roughly 105 of 300 genuinely poor-quality samples are missed. Precision on the Poor class is conversely very high at 0.98. Our model is thus conservative in the wrong direction for this application: when it declares water poor it is almost always correct, but it fails to raise an alarm in a third of poor-quality cases. Fig -6 visualises the resulting confusion structure.

Table -5: Per-class results, Gradient Boosting

Class	Prec.	Rec.	F1	Support
Excellent	0.83	0.99	0.91	280
Good	0.85	1.00	0.91	280
Macro avg.	0.89	0.88	0.87	860

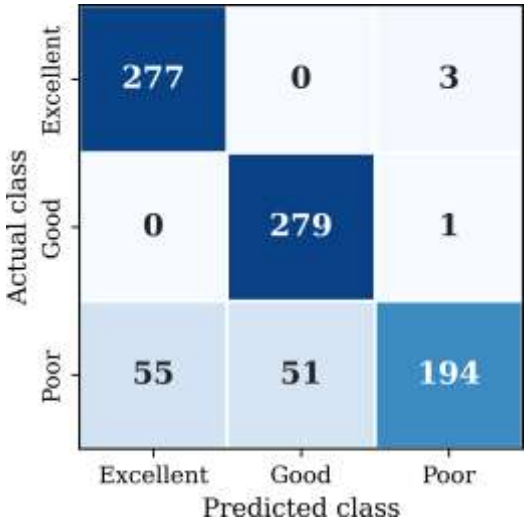


Fig -6: Confusion matrix for Gradient Boosting on the held-out test set. A dominant error mode is misclassification of Poor samples as Excellent or Good.

5.2 Feature-Set Ablation

Table -6 is the central result of this paper. It isolates the classification penalty attributable to restricting the model to microcontroller-measurable inputs, holding the classifier and protocol fixed.

Restricting the model to three deployable sensors costs 0.1198 macro F1 relative to the full 14-parameter panel, a relative reduction of 12.1%. We regard this as the honest statement of the trade-off: a system requiring no reagents, no titration and no trained operator retains roughly seven-eighths of the discriminative performance of full laboratory assessment on this dataset.

A second row deserves emphasis. Adding alkalinity and hardness lifts F1 from 0.8664 to 0.9035 — a substantial 3.7-point gain. It would be tempting to include them and report the higher figure, and we note that doing so would be methodologically indefensible for a paper claiming low-cost deployment, since both require titration. We report the row explicitly to make the temptation visible and to quantify what is being declined.

Table -6: Feature-set ablation (Gradient Boosting)

Feature set	n	Acc.	F1
T + pH + Turbidity (proposed)	3	0.8721	0.8664
Proposed + Alkalinity, Hardness	5	0.9058	0.9035
Laboratory parameters only	11	0.9326	0.9319
All parameters (upper bound)	14	0.9860	0.9861

A first row is deployable on an ESP32 without reagents. Rows two to four require wet-chemistry instrumentation to varying degrees.

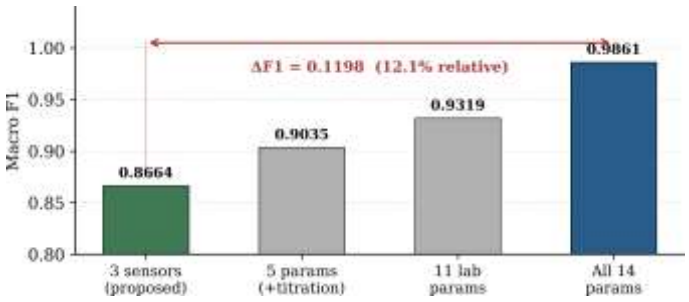


Fig -7: Cost–accuracy trade-off. A green bar marks the only configuration deployable without reagents or titration; the bracketed interval is the macro F1 forfeited by that restriction.

5.3 Soft-Sensing Regression: A Negative Result

We attempted to predict DO and BOD from the three deployable inputs. Table -7 reports the outcome. DO prediction essentially fails: the best model reaches $R^2 = 0.1536$, and the held-out R^2 for Random Forest is 0.0585. BOD is predicted more successfully at $R^2 = 0.4447$, but not to a standard that would substitute for measurement.

Table -7: Soft-sensing regression from three sensors

Target / Model	RMSE	MAE	R^2
DO — Gradient Boosting	1.6701	1.0405	0.1536
DO — Linear Regression	1.7108	1.2101	0.1118
DO — Random Forest	1.7614	1.1012	0.0585
BOD — Gradient Boosting	1.7553	1.1038	0.4447
BOD — SVR (RBF)	1.7953	1.0674	0.4191
BOD — Random Forest	1.8178	1.1329	0.4044

Section 6 argues that this result reflects the dataset rather than the sensors. Fig -8 summarises the outcome across models. Turbidity is the only deployable input with appreciable correlation to either target ($r = -0.35$ with DO, -0.28 with BOD); temperature and pH contribute almost nothing.

5.4 Edge Deployment

A three-layer fully connected network (32–16–3 units, 707 parameters) was trained on the deployable inputs, reaching 84.07% test accuracy — below Gradient Boosting, as expected from [6]. It is retained because tree ensembles are less convenient to deploy through TensorFlow Lite Micro. Fig -9 shows the training history.

A quantized model occupies 4.37 KB, or 0.107% of the 4 MB flash typical of an ESP32 module, and fits comfortably within the roughly 320 KB of usable RAM. Deployment is therefore not memory-constrained by a wide margin.

We deliberately do not claim quantization as a compression success. An INT8 to float32 size ratio is only 1.10×, because a 707-parameter model is dominated by FlatBuffer graph metadata rather than by weights. A larger apparent reduction from 33.22 KB to 4.81 KB is an artifact of Keras container overhead, not of any optimisation. That 0.35 percentage point accuracy difference in favour of INT8 is likewise within single-split noise and is not interpreted as a benefit of quantization.

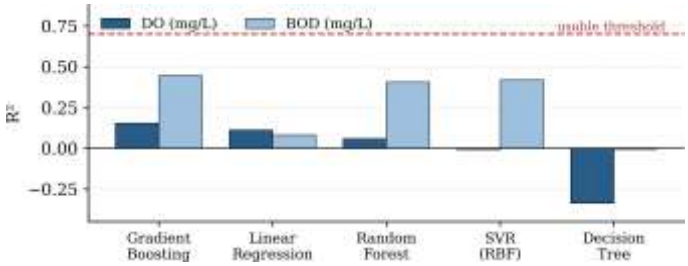


Fig -8: Coefficient of determination for soft-sensing from the three deployable inputs. No configuration approaches a usable standard for DO; BOD is predicted better but still inadequately. Negative R^2 indicates performance worse than predicting the mean.

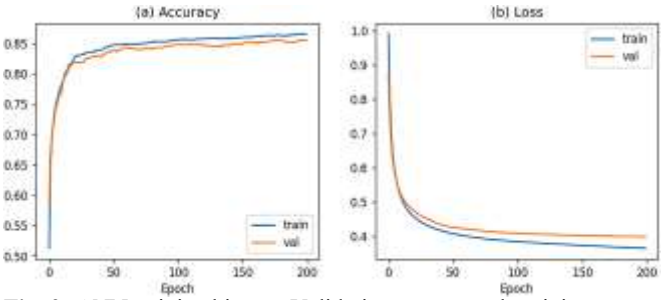


Fig -9: ANN training history. Validation curves track training curves without divergence, indicating the 707-parameter model does not overfit 3,440 samples.

5.5 Comparison with Reported Deployments

Table -9 situates the proposed configuration against the sensor complements of published IoT aquaculture systems. Our comparison is limited to instrumentation and scale rather than accuracy, since the datasets differ in target variable, class definition and collection methodology, and reported accuracies across them are not commensurable. What the table establishes is that the three-parameter configuration evaluated here is not an artificial restriction imposed for this study but the modal configuration of deployed systems in the literature.

One difference is worth noting. Field datasets in rows one to four sample densely in time from a small number of ponds, so consecutive records are strongly autocorrelated and an independent-and-identically-distributed train/test split would overstate generalisation. Our dataset has no temporal structure at all, which removes that particular hazard but introduces the different one discussed in Section 6.2. Neither construction is unambiguously preferable; they fail in different directions, and a study seeking to establish field performance would need both.

6. DISCUSSION AND LIMITATIONS

6.1 Why Soft-Sensing Failed

A natural reading of Table -7 is that temperature, pH and turbidity are inadequate predictors of dissolved oxygen. We believe this reading is wrong, and that stating it would propagate an incorrect conclusion into the literature.



Fig -10: Artifact size across representations. A large reduction from the Keras container to TFLite reflects container overhead rather than optimisation; the further INT8 reduction is only 1.10×.

Table -8: Edge deployment characteristics

Representation	Acc.	Size (KB)	Latency (ms)
Keras float32	0.8407	33.22	—
TFLite float32	0.8407	4.81	0.0021
TFLite INT8	0.8442	4.37	0.0020

Latency measured on an x86 host CPU, not on target hardware; see Section 6.

Table -9: Sensor complements in reported IoT aquaculture systems

Work	Controller	Parameters	Records
Islam & Kashem [1]	Arduino	pH, temp., turbidity	40,280
Siswanto et al. [2]	ESP8266	pH, TDS, temp.	118,286
Udanor et al. [3]	Micro-controller	Multi-parameter set	—
Baena-Navarro [7], [8]	IoT node	DO, pH, temp., turbid-ity	6 mo. hourly
This work	ESP32	Temp., pH, turbidity	4,300

Sample counts are not directly comparable: field studies log continuously from few ponds, whereas the dataset used here is class-balanced by construction.

Dissolved oxygen saturation is governed by Henry's law and falls monotonically with temperature; in field data the two are strongly negatively correlated. Our dataset exhibits $r = +0.027$ pooled, and within-class correlations of $+0.025$, -0.013 and $+0.031$. A physical relationship is simply absent. Combined with the plausibility violations in Table -1 — temperatures to 84.25 °C, pH values above the scale maximum — evidence indicates that parameters were drawn independently within class-conditional ranges. A regressor cannot recover a relationship that was never encoded. That failure is a property of this dataset, not a measurement of what low-cost sensors can achieve, and we recommend that soft-sensing feasibility be assessed only on field-collected data such as [1], [7] or [8].

6.2 Circularity of the Classification Task

A generative procedure has a subtler consequence for the classification result. Because labels were assigned from threshold values on the very features used as inputs [4], the classification task is partially circular: the model recovers a rule that was imposed rather than a relationship that was observed. That 87.21% figure should therefore be read as an upper-bound estimate of performance on threshold-consistent data, not as a prediction of accuracy in a pond. Our ablation in Table -6 is more robust to this concern than the absolute accuracy is, because circularity affects all four rows comparably and largely cancels in the comparison.

6.3 Latency Measurements

Table -8 reports sub-microsecond inference latencies measured on an x86 host. These do not transfer to an ESP32. Float32 kernels on the host benefit from XNNPACK vectorisation while INT8 kernels receive no comparable acceleration, which is why quantization yields no host-side speedup. On a microcontroller without a floating-point unit the ordering would likely reverse. We therefore rest the deployment argument on memory footprint, which is a property of the artifact rather than of the benchmarking host, and treat on-target latency as unmeasured.

6.4 Operational Implication of Poor-Class Recall

A recall of 0.65 on the Poor class means roughly one in three water quality failures goes undetected. For an early-warning system this is the error that matters, since a missed alarm can precipitate a fish kill while a false alarm costs only an inspection. Future work should adjust the decision threshold or apply class weighting to trade the currently very high Poor-class precision of 0.98 for improved recall.

6.5 Absence of Hardware Validation

Our most significant limitation is that no measurements from physical sensors appear in this study. Architecture of Fig -1 is specified but not instrumented, and the reported accuracy therefore characterises a model rather than a system. Sensor drift, analog-to-digital converter noise, temperature cross-sensitivity of pH probes and fouling of optical turbidity sensors are all absent from the evaluation and would each be expected to degrade field performance.

7. CONCLUSIONS

We quantified the accuracy cost of restricting aquaculture water quality classification to the three parameters a microcontroller can measure. Gradient Boosting achieves 87.21% accuracy and 0.8664 macro F1 from temperature, pH and turbidity alone, sacrificing 0.1198 F1 — 12.1% relative — against a full 14-parameter laboratory panel, with stability confirmed across ten random splits. A quantized inference artifact occupies 4.37 KB, well within ESP32 constraints. We also report and diagnose a negative soft-sensing result, attributing it to the synthetic construction of the dataset rather than to sensor limitations.

Three directions follow. First, and most importantly, the model should be validated against field measurements: even fifty readings from a deployed node, labelled against a reference laboratory assay or a published potability standard [20], [21], would convert this from a modelling study into a system evaluation. Second, the same ablation should be repeated on field-collected datasets such as [1], [7] and [8], where physical relationships between parameters are preserved and soft-sensing may well succeed. Third, on-target latency and energy per inference should be measured on ESP32 hardware, which would allow the duty-cycling and battery-life analysis that a solar-powered deployment requires.

REFERENCES

1. M. M. Islam and M. A. Kashem, “Real-time dataset of pond water for fish farming using IoT devices,” *Data in Brief*, vol. 51, art. 109761, Dec. 2023, doi: 10.1016/j.dib.2023.109761.
2. B. Siswanto, Y. Dani, D. Morika and B. Mardiyana, “A simple dataset of water quality on aquaponic fish ponds based on an internet of things measurement device,” *Data in Brief*, vol. 48, art. 109248, Jun. 2023, doi: 10.1016/j.dib.2023.109248.
3. C. N. Udanor, N. I. Ossai, E. O. Nweke, B. O. Ogbuokiri, A. H. Eneh, C. H. Ugwuishiwu, S. O. Aneke, A. O. Ezuwgu, P. O. Ugwoke and A. Christiana, “An internet of things labelled dataset for aquaponics fish pond water quality monitoring system,” *Data in Brief*, vol. 43, art. 108400, Aug. 2022, doi: 10.1016/j.dib.2022.108400.
4. V. Veeramsetty, R. Arabelli and T. Bernatin, “Aquaculture — Water Quality Dataset,” *Mendeley Data*, V1, 2024, doi: 10.17632/y78ty2g293.1.
5. R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang and P. Warden, “TensorFlow Lite Micro: Embedded machine learning for TinyML systems,” in *Proc. Machine Learning and Systems (MLSys)*, vol. 3, 2021, pp. 800–811.
6. L. Grinsztajn, E. Oyallon and G. Varoquaux, “Why do tree-based models still outperform deep learning on typical tabular data?,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, Datasets and Benchmarks Track, New Orleans, LA, USA, 2022.
7. R. Baena-Navarro, Y. Carriazo-Regino and F. Torres-Hoyos, “IoT monitoring dataset of water quality and tilapia (*Oreochromis niloticus*) health in aquaculture ponds in Montería, Colombia,” *Mendeley Data*, V1, 2024, doi: 10.17632/3g2b4sh65m.1.
8. R. Baena-Navarro, Y. Carriazo-Regino, F. Torres-Hoyos and J. Pinedo-López, “Water quality monitoring dataset for tilapia aquaculture in Montería, Colombia,” *Mendeley Data*, V1, 2024, doi: 10.17632/dgdr2kfbyt.1.
9. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam and D. Kalenichenko, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 2704–2713.
10. J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
11. L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
12. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
13. C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
14. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
15. M. Abadi et al., “TensorFlow: A system for large-scale machine learning,” in *Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, Savannah, GA, USA, 2016, pp. 265–283.
16. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
17. C. R. Banbury et al., “MLPerf Tiny Benchmark,” in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, Datasets and Benchmarks Track, 2021.
18. P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA, USA: O’Reilly Media, 2019.
19. C. E. Boyd, *Water Quality: An Introduction*, 3rd ed. Cham, Switzerland: Springer, 2020.
20. World Health Organization, *Guidelines for Drinking-water Quality*, 4th ed. incorporating the 1st addendum. Geneva, Switzerland: WHO, 2017.
21. Bureau of Indian Standards, *IS 10500:2012 — Drinking Water Specification*, 2nd rev. New Delhi, India: BIS, 2012.
22. J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, May 2007.