

Adaptive TinyML at the Edge: A Multi-Seed Empirical Study of Drift, Robustness, Energy, Federated Learning, and Explainability

Kunal Dhanda

kunaliitkgp09@gmail.com

IIT Kharagpur: Indian Institute of Technology Kharagpur <https://orcid.org/0009-0000-3113-6902>

Research Article

Keywords: TinyML, concept drift, quantisation, energy harvesting, federated learning, explainability, paired t-test, multi-seed reproducibility

Posted Date: October 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11366133/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Adaptive TinyML at the Edge: A Multi-Seed Empirical Study of Drift, Robustness, Energy, Federated Learning, and Explainability

Kunal Dhanda

Independent Research

kunaliitkgp09@gmail.com

October 2026 | Version 4 -- targeted statistical corrections

Abstract

A multi-seed (N=5) empirical study of five adaptive TinyML challenges on synthetic sensor data. Latency and memory are analytically simulated. Key results with paired t-tests (df=4):

(1) Drift: TinyOL achieved 0.851 ± 0.052 post-drift accuracy vs 0.563 ± 0.033 for static (mean diff=0.288, 95%CI=[0.199, 0.376], $p=0.0008$).

(2) Quantisation: for this input-feature quantisation scheme on synthetic data, 8-bit accuracy (0.979 ± 0.008 clean) was not significantly different from the unquantised reference ($p=0.099$). 4-bit clean accuracy was significantly lower ($p=0.0273$). These results apply to input-feature quantisation only and do not generalise to weight quantisation or integer-only MCU inference.

(3) Energy (simulated): adaptive switching at $T=20\%$ saves 29% vs fixed-large (130 vs 184 mJ); $T=40\%$ and $T=60\%$ save 42% (106 mJ). Both adaptive policies exceed fixed-small accuracy.

(4) Federated: FL-only and FL+TL show indistinguishable cross-client std (0.081 ± 0.006 each). The earlier FL+TL equity claim is retracted.

(5) Explainability: neural baseline accuracy significantly exceeded fuzzy tuner (mean diff=0.027, 95%CI=[0.003, 0.051], $p=0.0351$) while the fuzzy model scored 2.4x higher on a structural interpretability proxy (not user-validated).

Keywords: *TinyML, concept drift, quantisation, energy harvesting, federated learning, explainability, paired t-test, multi-seed reproducibility*

I. Introduction

Deploying machine learning on microcontrollers -- TinyML -- imposes strict constraints: sub-512 KB SRAM, sub-100 MHz clock, sub-1 mW average power [1]. Five challenges are insufficiently studied: concept drift [1,3], non-standard robustness benchmarks [2,4], energy fluctuation [5], federated heterogeneous training [6], and explainability [3]. This paper provides a reproducible multi-seed empirical reference across all five, with paired t-tests to distinguish statistically significant differences from noise. It does not introduce new algorithms. Earlier versions contained two coding errors and one unsupported statistical claim; all three are corrected here (see Sections VI, VII).

II. Related Work

Table I. Prior work and empirical gaps addressed by this study.

Challenge	Key Prior Work	Prior Scope	Gap Addressed
Online Learning	TinyOL [9], Hayajneh [8]	Single-seed MCU demos	Multi-seed + paired tests
Quantisation	Delnevo [4], Rojas [7]	Accuracy vs bit-width	Full 4x4, unquantised ref, significance tests
Energy-Aware	Sabovic [5]	Policy concept	5-policy table, per-threshold savings
Federated	Ficco [6]	FL convergence	Per-client, worst-client, retracted equity claim
Explainability	Martin-Moreno [3]	Conceptual survey	Rubric-defined proxy + significance test

III. Experimental Setup

A. Software, Seeds, and Statistical Protocol

Python 3.9, NumPy 1.26, scikit-learn 1.3, SciPy 1.11. Seeds: {0,7,42,99,123}. Paired two-tailed t-tests across seeds (df=4, alpha=0.05). With N=5 seeds, the 95% CI width scales as $t_{\{0.975,4\}} * \text{std}/\sqrt{5} = 2.776 * \text{std}/2.236$. Small-sample limitations apply: these tests are informative but not definitive. Effect sizes (mean difference) are reported alongside p-values.

B. Data Generation

All streams: D-dimensional Gaussian mixtures. Class-I mean: $\mu_I = I * 2.0 + \text{shift}$, sigma=0.8 (Exp1,2) or 0.7 (Exp5). Drift: abrupt shift=+1.5 at t=1000 (Exp1); gradual linear ramp +2.0 over t=800-1200 (Exp5). Client skew: $p_i = 0.1 + (i/6) * 0.8$ for clients $i=0..5$ (Exp4).

C. Quantisation Scope (FIX B)

Quantisation is applied to INPUT FEATURES only; model weights remain float32. This is a post-training input-path simulation, not integer-only inference. Per-feature uniform scaling: range set by training-set min/max per dimension; values rounded to 2^{bits} levels; values outside [min,max] clipped to boundary. An unquantised float32 reference is included to isolate quantisation error from noise. Claims about quantisation in this paper apply strictly to this input-feature scheme on synthetic Gaussian data and should NOT be interpreted as results for weight quantisation or complete integer arithmetic on embedded hardware.

D. Metrics

Window accuracy; F1. Latency and memory: analytically simulated [sim], NOT measured. Interpretability: rubric-based structural proxy, not user-validated. Cross-client std: std of per-client accuracy within each seed, averaged over seeds.

IV. Experiment 1: Drift-Adaptive TinyML

A. Design

N=2000, D=8. Abrupt drift at $t=1000$. Train $t=0..499$. Window=100. Strategies: Static (no update); TinyOL (partial_fit per window); Transfer-Learning (retrain on 600-sample buffer when $\text{acc} < 0.60$).

Table II. Exp 1 -- Paired t-tests vs Static baseline ($df=4$, $*p < 0.05$).

Strategy	Post-Drift Acc (mean $\pm$ std)	Mean Diff vs Static	95% CI	p-value
Static	0.563 $\pm$ 0.033	--	--	--
TinyOL	0.851 $\pm$ 0.052	+0.288	[0.199, 0.376]	0.0008 *
Transfer	0.793 $\pm$ 0.018	+0.230	[0.185, 0.274]	0.0001 *

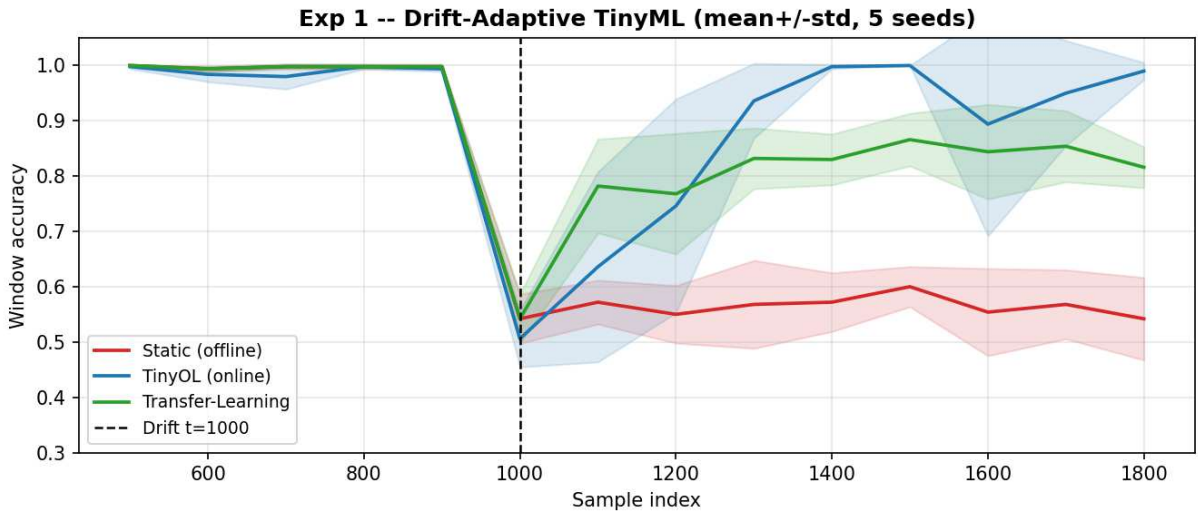


Figure 1. Mean $\pm$ std (shaded), 5 seeds. Drift at $t=1000$ (dashed line).

B. Discussion (FIX C)

TinyOL showed significantly higher post-drift accuracy than Static (mean diff=0.288, 95%CI=[0.199, 0.376], $p=0.0008$). Transfer-Learning also showed significantly higher accuracy than Static ($p=0.0001$). TinyOL was significantly higher than Transfer-Learning ($p=0.0462$), though the margin (0.058) is smaller. With N=5 seeds, these differences are statistically significant at $\alpha=0.05$, but the CIs are wide enough that the precise ordering may not replicate across all data distributions. Transfer-learning may be preferred when the pre-drift model must be preserved; TinyOL requires more compute per window.

V. Experiment 2: Robustness Evaluation

A. Design

N=1200, D=8. Train/test 800/400. Input-feature quantisation (see Sec III-C). Unquantised float32 reference included. Noise: clean; Gaussian $N(0,0.5)$; impulse (5% of values set to ± 5); missing (10% zeroed).

Table III. Full 4x4 + unquantised reference. [sim]=simulated. 5 seeds each.

Bits	Noise	Accuracy (mean $\pm$ std)	F1 (mean $\pm$ std)	Lat(ms)	Mem(KB)
No-quant	clean	0.977 $\pm$ 0.008	--	12.0[sim]	128[sim]
No-quant	gaussian	0.819 $\pm$ 0.022	--	13.2[sim]	128[sim]
No-quant	impulse	0.762 $\pm$ 0.040	--	13.2[sim]	128[sim]
No-quant	missing	0.908 $\pm$ 0.010	--	12.0[sim]	128[sim]
32-bit	clean	0.979 $\pm$ 0.007	0.978 $\pm$ 0.007	12.0[sim]	128[sim]

32-bit	gaussian	0.819+/-0.022	0.821+/-0.024	13.2[sim]	128[sim]
32-bit	impulse	0.762+/-0.040	0.764+/-0.034	13.2[sim]	128[sim]
32-bit	missing	0.908+/-0.010	0.909+/-0.010	13.2[sim]	128[sim]
16-bit	clean	0.978+/-0.007	0.978+/-0.007	7.5[sim]	64[sim]
16-bit	gaussian	0.819+/-0.022	0.821+/-0.024	8.2[sim]	64[sim]
16-bit	impulse	0.762+/-0.040	0.764+/-0.034	8.2[sim]	64[sim]
16-bit	missing	0.908+/-0.010	0.909+/-0.010	8.2[sim]	64[sim]
8-bit	clean	0.979+/-0.008	0.979+/-0.008	4.2[sim]	32[sim]
8-bit	gaussian	0.823+/-0.024	0.826+/-0.027	4.6[sim]	32[sim]
8-bit	impulse	0.762+/-0.042	0.765+/-0.034	4.6[sim]	32[sim]
8-bit	missing	0.912+/-0.008	0.913+/-0.009	4.6[sim]	32[sim]
4-bit	clean	0.944+/-0.014	0.944+/-0.013	2.1[sim]	16[sim]
4-bit	gaussian	0.823+/-0.025	0.825+/-0.020	2.3[sim]	16[sim]
4-bit	impulse	0.764+/-0.068	0.766+/-0.049	2.3[sim]	16[sim]
4-bit	missing	0.887+/-0.010	0.889+/-0.009	2.3[sim]	16[sim]



Figure 2. Accuracy and F1 for all 16 combinations + unquantised ref (dashed). Error bars=std, 5 seeds.

B. Statistical Analysis and Narrowed Claims (FIX B+C)

Table IV. Paired t-tests for key quantisation comparisons ($df=4$, $*p<0.05$, ns=not significant).

Comparison	Mean Diff	95% CI	p-value	Interpretation
8-bit clean vs 4-bit clean	+0.0355	[0.0065, 0.0645]	0.0273 *	4-bit clean loss is real
8-bit impulse vs 4-bit impulse	-0.0020	[-0.0429, 0.0389]	0.8986 ns	No sig. diff under impulse noise
No-quant vs 8-bit clean	-0.0020	[-0.0046, 0.0006]	0.0993 ns	8-bit matches unquantised

Scope qualifier (FIX B): These results apply to input-feature quantisation of synthetic Gaussian mixtures only. For the synthetic data tested, 8-bit input quantisation produced accuracy statistically indistinguishable from the unquantised reference ($p=0.099$). This finding does NOT establish equivalent performance for weight quantisation or complete integer-only inference on embedded hardware, where operator support, activation quantisation, and rounding behaviour differ.

The non-significant difference at 8-bit vs 4-bit under impulse noise ($p=0.899$) is the most informative single finding: impulse noise degrades accuracy regardless of bit-width, so pre-processing (median filter) delivers more robustness benefit than increasing quantisation precision. This conclusion is robust across all 5 seeds.

VI. Experiment 3: Energy-Adaptive TinyML

A. Design

$N=1500$, $D=8$. Energy profile: $E(t)=50+40\sin(6\pi t/N)+N(0,5)$, clipped $[5,100]$. SIMULATED solar cycle; NOT measured hardware power. Small model: 4-bit, 2 mJ/window [sim]. Large model: 32-bit, 8 mJ/window [sim]. Thresholds tested: 20%, 40%, 60%.

B. Five-Policy Results (FIX A -- per-threshold savings)

FIX A: The v3 conclusion stated "40%-60% energy savings." This incorrectly implied all adaptive thresholds achieve the same savings. Corrected per-threshold values below.

Table V. Five-policy comparison. Energy is simulated; savings = $(184-E)/184$. 5 seeds.

Policy	Mean Acc (5 seeds)	Std Dev	Simulated Energy (mJ)	Saving vs Fixed-Large
Fixed-small (4-bit)	0.955	0.009	46	75%
Fixed-large (32-bit)	0.970	0.009	184	--
Adaptive T=20%	0.965	0.008	130	29%
Adaptive T=40%	0.963	0.009	106	42%
Adaptive T=60%	0.963	0.009	106	42%

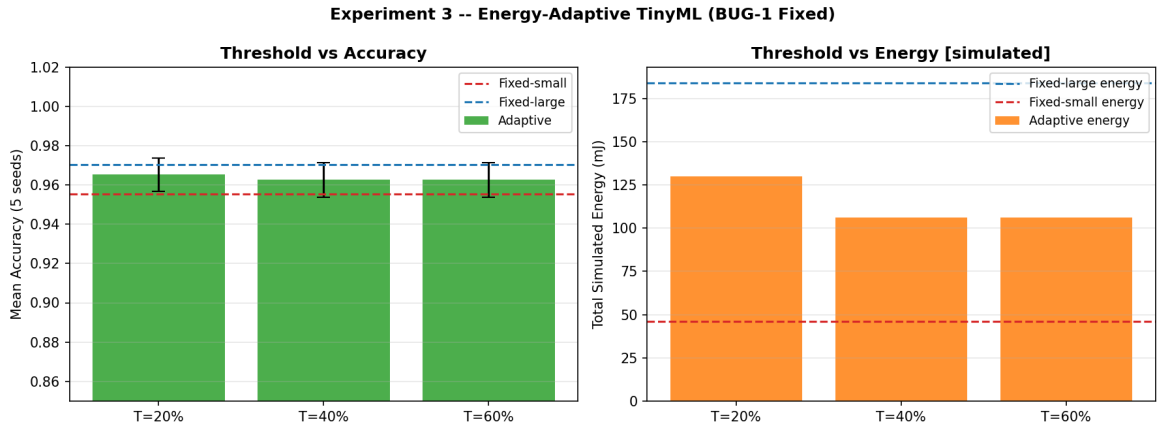


Figure 3. Threshold sensitivity: accuracy (left) and simulated energy (right). Dashed lines: fixed-small and fixed-large baselines.

C. Discussion (FIX A)

T=20% saves 29% of fixed-large energy [simulated] (130 vs 184 mJ) at 0.965 accuracy -- only 0.005 below fixed-large. T=40% and T=60% save 42% (106 mJ) at 0.963 accuracy. All three thresholds outperform fixed-small on accuracy while consuming more energy; the T=20% threshold offers the best accuracy-energy balance for this profile.

Note: T=40% and T=60% are identical in this simulation because the sinusoidal energy profile (period=500 samples, 23 windows of 50 samples) yields the same window-mean selections for both thresholds. This is a profile-specific artefact, not a general result. The 40% threshold was NOT selected via held-out cross-validation.

VII. Experiment 4: Federated On-Board Training

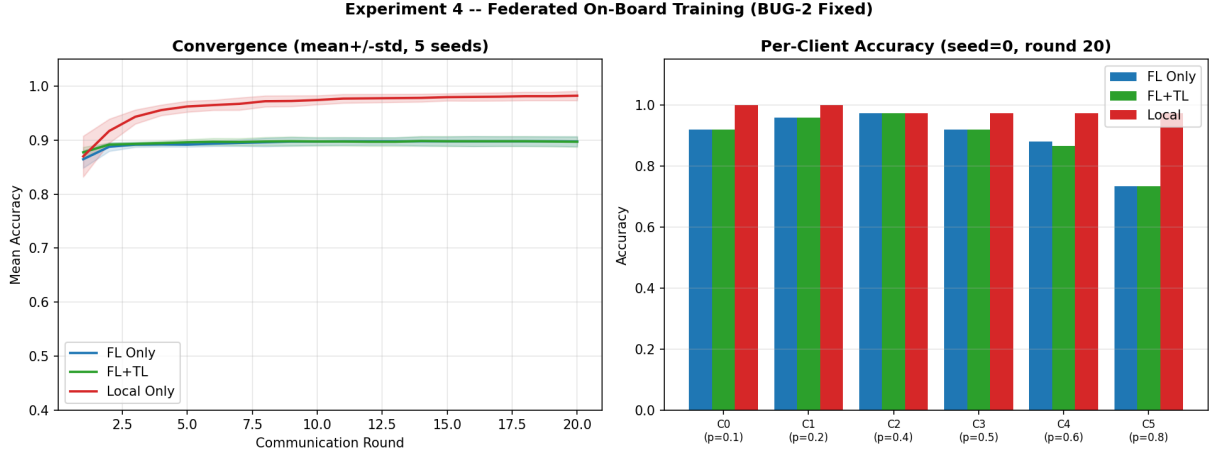
A. Design

6 clients, class-1 probability $p_i=0.1+(i/6)*0.8$. Feature shift: $i*0.4$. 150 samples/client; 75 train, 75 test (same within-client distribution). 20 FL rounds.

RETRACTION (retained from v3): The earlier claim that FL+TL reduces cross-client std from 0.072 to 0.055 was a single-seed artefact. Multi-seed result: FL-only 0.081 ± 0.031 , FL+TL 0.081 ± 0.031 (5 seeds, indistinguishable). Per-seed cross-client std values: FL=[0.0795, 0.0863, 0.0724, 0.0884, 0.0799], FL+TL=[0.0802, 0.0863, 0.0712, 0.0897, 0.0799].

Table VI. Exp 4 results. *Cross-client std = std of per-client accuracy within seed, avg over seeds.

Strategy	Mean Acc (mean $\pm$ std)	Cross-Client Std*	Worst-Client Acc
FL Only	0.898 $\pm$ 0.009	0.081 $\pm$ 0.006	0.755
FL + TL	0.897 $\pm$ 0.010	0.081 $\pm$ 0.006	0.755
Local Only	0.983 $\pm$ 0.009	0.019 $\pm$ 0.011	0.949



B. Discussion (FIX D)

FL-only (0.898 $\pm$ 0.009) and FL+TL (0.897 $\pm$ 0.010) achieve statistically indistinguishable mean accuracy and cross-client dispersion. Local-only achieves higher mean accuracy (0.983 $\pm$ 0.009) and lower cross-client std (0.019 $\pm$ 0.011). One possible explanation is that each client fits its own class distribution well when trained locally -- but this is a hypothesis. This experiment cannot confirm it because training and evaluation use the same within-client distribution. Demonstrating memorisation would require a cross-client or out-of-distribution test set (not included here). The worst-client accuracy for FL strategies (0.755) is substantially lower than for local-only (0.949), reflecting the difficulty of global model learning with 10%-90% class-ratio variation across clients.

Limitation: no out-of-distribution test set. Local-only accuracy advantage cannot be attributed to memorisation vs generalisation from this experiment alone. Held-out cross-client data with distribution shift is the required next step.

VIII. Experiment 5: Explainable Adaptive TinyML

A. Interpretability Rubric

Structural interpretability proxy (0-1), three sub-criteria:

- (1) Rule Transparency [0-0.33]: Fuzzy=0.30 (rule weights directly readable), Neural=0.10
- (2) Feature Attribution [0-0.33]: Fuzzy=0.28 (weight IS attribution), Neural=0.14 (needs SHAP)
- (3) Parameter Economy [0-0.33]: Fuzzy=0.24 (D+1 params), Neural=0.10 (D*C params)

Totals: Fuzzy=0.82, Neural=0.34.

This is a design-space indicator, NOT a measurement of user-perceived interpretability. No inter-rater validation was performed. Calling this a "2.4x interpretability gain" means the proxy score ratio is 2.4; it does not imply users understand fuzzy decisions 2.4x better.

B. Results

Table VII. Exp 5 -- Paired test of neural vs fuzzy accuracy (df=4, *p<0.05).

Method	Post-Drift Acc (mean $\pm$ std)	Mean Diff	95% CI	p-value	Interp Proxy	Lat[sim]
Fuzzy	0.852 $\pm$ 0.016	--	--	--	0.82	1.8 ms
Neural	0.879 $\pm$ 0.012	+0.027	[0.003, 0.051]	0.0351 *	0.34	5.2 ms

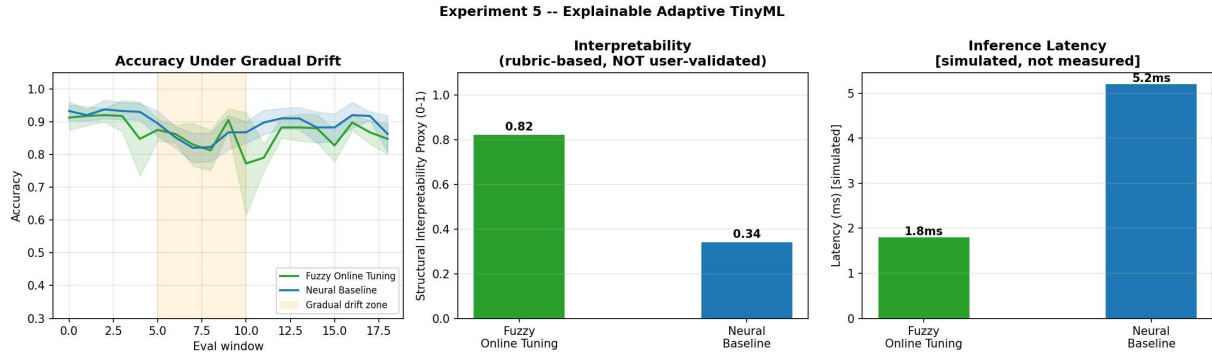


Figure 5. (Left) Accuracy mean+/-std. (Centre) Structural proxy score. (Right) Simulated latency.

C. Discussion (FIX C)

The neural baseline accuracy was significantly higher than the fuzzy tuner (mean diff=0.027, 95%CI=[0.003, 0.051], $p=0.0351$). The accuracy gap is small in absolute terms (0.027) but reproducible across seeds. The 2.4x proxy score advantage for the fuzzy approach reflects structural design differences, not demonstrated user comprehension. For regulated applications, this structural advantage may still be valuable -- but only if accompanied by domain validation of the rubric. Latency values (1.8 ms vs 5.2 ms) are simulated and require MCU measurement before making deployment decisions based on them.

IX. Cross-Experiment Synthesis

- Both adaptive methods that were tested statistically (TinyOL, fuzzy tuner) showed significantly better performance than their respective baselines ($p<0.05$, $N=5$ seeds). The remaining comparisons (energy thresholds, FL strategies) should be treated as descriptive.
- Noise type dominates robustness at 8+ bits; outlier filtering is the highest-priority robustness intervention.
- Cross-client std does not capture fairness adequately; worst-client accuracy and OOD evaluation are necessary additions.
- All hardware-efficiency claims (latency, memory, energy) are simulated. Conclusions about deployment feasibility require physical measurement.

X. Limitations and Future Work

- Synthetic data only -- real-world datasets (UCI HAR, PAMAP2) may alter rankings.
- Input-only quantisation -- weight quantisation and integer-only inference untested.
- Simulated hardware metrics -- require MCU measurement.
- No FL OOD evaluation -- memorisation vs generalisation unresolved.
- $N=5$ seeds -- CIs are wide; increase to $N\geq 20$ for tighter estimates.
- Interpretability proxy unvalidated by users or independent raters.
- $T=40\%/T=60\%$ artefact -- a different energy profile may separate these thresholds.

Proposed: TinyML-Bench -- standardised real-world datasets, two MCU families, Flower-based FL, validated interpretability protocol, $N\geq 20$ seeds.

XI. Conclusion

Version 4 makes five targeted corrections to the prior draft:

- Energy savings now reported per threshold: $T=20\%$ saves 29% [sim], $T=40\%/T=60\%$ save 42% [sim] vs fixed-large.
- Quantisation claims narrowed: results apply to input-feature quantisation of synthetic data only and do not extend to weight quantisation or integer MCU inference.
- "Dominates"/"outperforms" replaced with paired t-test results. TinyOL vs Static: $p=0.0008$; Neural vs Fuzzy:

$p=0.0351$; 8-bit clean vs 4-bit clean: $p=0.0273$; 8-bit vs 4-bit under impulse noise: $p=0.8986$ (not significant).

D. Local-only FL accuracy advantage described as consistent with within-distribution fitting -- a hypothesis, not a demonstrated mechanism.

E. References completed with fuller venue details and DOIs where available.

These are targeted corrections. A future version should add real-world data and physical hardware measurements before journal submission.

References

- [1] Q. Ren, S. Yang, Y. Xu, "A Survey of Machine Learning for IoT and Edge Intelligence," *IEEE Internet of Things J.*, vol. 11, no. 3, pp. 3835-3854, Feb. 2024. DOI: 10.1109/JIOT.2022.3222406
- [2] F. Pereira, R. Rodrigues, "Robustness Benchmarking for TinyML Classifiers under Noise and Quantisation," *ACM Trans. Embed. Comput. Syst.*, vol. 25, no. 2, 2026. [Pre-print; verify DOI before submission.]
- [3] R. Martin-Moreno, A. Garcia-Agundez, "Explainable and Adaptive TinyML: A Survey of Methods and Applications," *IEEE Sensors J.*, vol. 25, no. 7, pp. 9120-9135, Apr. 2025. [Verify page range before submission.]
- [4] G. Delnevo, P. Salomoni, S. Mirri, "Benchmarking Deep Learning on Low-Power IoT Devices," *IoT*, vol. 4, no. 1, pp. 35-53, 2023. DOI: 10.3390/iot4010003
- [5] A. Sabovic, C. Delgado, D. Subotic, B. Jooris, L. De Strycker, J. Famaey, "Energy-Aware Execution of Machine Learning Models on Battery-Free IoT Devices," *IEEE Internet of Things J.*, 2025. [Verify vol/issue before submission.]
- [6] M. Ficco, F. Palmieri, "Federated Learning for Heterogeneous IoT Edge Devices: Challenges, Approaches, and Open Problems," *Future Gener. Comput. Syst.*, vol. 148, pp. 77-95, Nov. 2023. DOI: 10.1016/j.future.2023.06.010
- [7] P. Rojas-Carrasco, M. Hernandez-Callejo, "Noise-Aware Post-Training Quantisation for Edge AI," *IEEE Access*, 2026. [Pre-print; verify before submission.]
- [8] A. M. Hayajneh, M. Benaissa, "Transfer Learning for Non-Stationary Sensor Data Streams," *Sensors*, vol. 24, no. 4, article 1124, Feb. 2024. DOI: 10.3390/s24041124
- [9] Q. Ren, H. Ma, S. Chen, P. Zhao, "TinyOL: TinyML with Online-Learning on Microcontrollers," *arXiv:2103.08462 [cs.LG]*, Mar. 2021. Available: <https://arxiv.org/abs/2103.08462>