

Research Article

Energy-Aware TinyML for Batteryless Fire Detection Nodes Using LoRaWAN and Ambient Energy Harvesting



Modhar A. Hammoudy¹,  | Mustafa S. Qassab^{1,*}, 
 1 Department of Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq

Received: 31 / 7 / 2026 | Revised: 19 / 8 / 2026 | Accepted: 3 / 9 / 2026 | Online: 22 / 9 / 2026

KEYWORDS

TinyML
 Batteryless IoT
 Ambient energy harvesting
 LoRaWAN
 Wildfire detection
 Intermittent computing
 Energy-neutral operation
 On-device inference

ABSTRACT

Batteryless wireless sensor nodes promise maintenance-free environmental monitoring at large scale. However, their supply is intermittent and scarce. This conflicts directly with the low-latency, high-reliability requirements of early fire detection. This paper addresses that conflict. We present EAF-TinyML, an energy-aware tiny machine-learning framework for batteryless fire-detection nodes. The framework combines ambient photovoltaic and thermoelectric harvesting, supercapacitor buffering, and a Long-Range Wide-Area Network (LoRaWAN) backhaul. Three mechanisms are co-designed around a single energy budget: a gated inference pipeline, a state-of-charge-driven adaptive duty-cycle controller, and a confidence-aware transmission policy. We formalise the node energy balance and derive the energy-neutrality condition that ties the duty-cycle period to the harvested power. We evaluate the framework on the public Smoke Detection multi-sensor dataset and on an ARM Cortex-M4 hardware-in-the-loop prototype driven by replayed harvesting traces. The quantised model occupies 41 kB of flash and runs one inference in 18 ms at 73 μ J. Detection F1 is 0.967 ± 0.006 over five-fold episode-wise cross-validation with three random seeds. Against two node-level analogues re-evaluated under a common energy model, EAF-TinyML reduces average energy per cycle by 38–61% and lowers median alarm latency to 9.4 s. Energy-neutral operation is sustained in bench-replayed traces down to about 250 lx of sustained illumination; below roughly 120 lx for more than 36 h, the node degrades gracefully to intermittent operation rather than failing. The results indicate that system-level energy co-design, not smaller models alone, makes self-powered fire sentinels practical. Field validation remains future work.

1. INTRODUCTION

Uncontrolled fires inflict severe ecological, economic, and human costs, and their damage grows super-linearly with the time between ignition and intervention. Consequently, a monitoring system's value depends less on peak sensitivity than on how early and reliably it raises an alarm across vast, often inaccessible areas. Dense networks of low-cost wireless sensors can in principle provide minute-scale detection over forests, peri-urban interfaces, and industrial sites, and recent deployments confirm both the demand and the commercial momentum behind this approach [1], [2], [3].

*Corresponding Author (Email): mustafa.qassab@uomosul.edu.iq

However, the dominant obstacle to large-scale deployment is energy. Battery-powered nodes impose a recurring maintenance burden that scales with network size; for a network of thousands of nodes distributed over rugged terrain, periodic battery replacement is economically and logistically prohibitive [4], [5]. Two converging technologies offer a path around this barrier. First, ambient energy harvesting paired with supercapacitor buffering enables batteryless nodes that operate intermittently as energy is available, eliminating chemical storage and its maintenance, replacement logistics, and disposal costs [6], [7], [8]. Removing the battery is therefore not only an operational convenience but a step towards sustainable, long-lived IoT infrastructure, since the environmental footprint of a dense sensing network is dominated by the cells it consumes over its lifetime. Second, tiny machine learning (TinyML) brings neural-network inference to milliwatt-class microcontrollers (MCUs), allowing nodes to discriminate genuine fire signatures locally instead of streaming raw data to the cloud [9], [10], [11], [12], [13].

These technologies are, unfortunately, in tension. Early fire detection demands low and bounded latency and high reliability, whereas batteryless operation is intrinsically intermittent, unpredictable, and energy-scarce [6], [14]. A node that naively runs a neural network on every wake cycle may exhaust its supercapacitor before it can transmit; one that transmits on every positive may saturate the LoRaWAN duty-cycle budget and drain its buffer. The central research problem this paper addresses is therefore: how can a self-powered node deliver timely, reliable fire alarms without violating its instantaneous and long-term energy constraints?

Positioning and scope of the claim. We state the boundaries of our novelty explicitly rather than leaving them to be inferred from Table 1. EAF-TinyML does not propose a new harvesting circuit, quantisation algorithm, or gating primitive; it uses each of these building blocks from established practice [21], [22], [27]. The closest prior work, the energy-aware TinyML study of Sabovic et al. [27], has already established that inference and communication on a battery-less device must be scheduled against the harvested-energy state. Our advance over that line of work is therefore evolutionary rather than revolutionary, and it lies in two specific places. The first is domain specialisation: the pre-screen statistics, the fusion of learned evidence with physical gas evidence, and the transmission economics are designed around the signal structure and the asymmetric false-alarm cost of fire sentinels rather than around generic classification. The second is a latency guarantee: we couple the energy-neutrality condition to an explicit bound on alarm latency through a clamped adaptive duty cycle, so energy scarcity degrades latency in a controlled, bounded way instead of degrading availability. This paper contributes the co-design of these known mechanisms into one energy budget, and the measured quantification of what that co-design buys, not any single new mechanism in isolation.

Contributions. Against this background, this paper contributes as follows.

1. A formal per-cycle energy balance for a harvesting fire-detection node, together with a closed-form energy-neutrality condition that ties the duty-cycle period to the harvested power (Section 3).
2. A two-stage gated inference pipeline in which an ultra-low-power statistical pre-screen wakes an 8-bit quantised 1D convolutional network only when the ambient signals are suspicious, cutting the expected inference energy by more than an order of magnitude (Section 4.1).
3. A state-of-charge-driven adaptive duty-cycle controller together with a confidence-aware LoRaWAN transmission policy, which jointly bounds alarm latency under scarce energy while preserving energy neutrality by construction (Sections 4.3 and 4.4).
4. An evaluation on a named public multi-sensor dataset and a Cortex-M4 hardware-in-the-loop prototype that reports cross-validated uncertainty, a sensitivity analysis of the three control parameters κ , τ and (w_1, w_2) , multi-day, seasonal and worst-case harvesting traces, and a safety-oriented error analysis under low state of charge (Sections 5 and 6).

The remainder of the paper is organised as follows. Section 2 reviews recent work on fire-detection sensor networks, TinyML, and batteryless IoT, and identifies the gap. Section 3 presents the system model and formalises the problem. Section 4 describes the proposed EAF-TinyML framework. Section 5 details the

experimental setup, and Section 6 reports the results and compares them with modern analogues. Section 7 concludes.

2. Related Work

2.1 IoT and WSN-Based Fire Detection

Wireless sensor networks have long been applied to fire detection. Dampage et al. combined a multi-hop WSN with a machine-learning regression model to flag incipient forest fires, establishing the multi-sensor (temperature, humidity, gas) feature paradigm still used today [15]. Subsequent work has improved network topology and connectivity: LoRa and LoRaWAN have become the de facto backhaul for remote monitoring because of their kilometre-scale range and milliwatt transmit budgets [1], [16], [17]. Brito et al. showed that on-node data filtering substantially reduces LoRaWAN traffic in a forest WSN [16], while De Rango et al. built a CO₂-centred smart-sensor early-warning network [18]. In parallel, vision-based detectors using YOLO-family models achieve strong smoke and flame recognition from ground and aerial imagery [19], and hybrid architectures combine camera vision with reinforcement-learning control for active monitoring [3]. Camera-based methods, however, demand power, line-of-sight, illumination, and broadband backhaul that are rarely available at a batteryless node [3].

2.2 TinyML on Microcontrollers

TinyML deploys quantised neural networks on MCUs with kilobytes of RAM and milliwatt power envelopes [9], [12], [13], with a dedicated line of work targeting low-power consumer edge devices [20]. Enabling techniques include post-training and quantisation-aware 8-bit integer quantisation, which shrinks models fourfold with minimal accuracy loss [21], and structured pruning, which removes redundant weights to fit tight memory budgets [22]. Practical pipelines built on TensorFlow Lite Micro and CMSIS-NN now run convolutional models for keyword spotting, anomaly detection, and object detection on Cortex-M devices [23], [24], and application studies now span domains from biomedical classification to consumer edge analytics [25]. Surveys emphasise that energy, not accuracy, is the binding constraint for always-on TinyML, and that careful benchmarking of energy per inference is essential [10], [26]. Yet most TinyML studies assume a stable supply and optimise accuracy or latency in isolation rather than under an intermittent energy budget.

2.3 Batteryless and Energy-Harvesting IoT

Batteryless devices replace chemical storage with a small capacitor charged by harvested energy, and therefore operate intermittently through repeated charge–discharge cycles [6]. Intermittent computing provides mechanisms—checkpointing, task-based execution—to make computation correct across power failures [14]. Energy-harvesting surveys catalogue photovoltaic, thermoelectric, RF, and kinetic sources and their cross-layer integration [7], [8], and energy-autonomous LoRaWAN nodes have been demonstrated for mass-scale IoT [5]. Most relevant to this work, Sabovic et al. studied energy-aware TinyML on battery-less devices and showed that inference and communication must be scheduled against the harvested-energy state [27]. Their work, however, targets generic classification and does not address the latency-critical, two-stage gating and transmission economics specific to fire sentinels.

Research gap. Table 1 contrasts representative recent systems along the three axes that matter here. No prior work simultaneously (a) runs on-device neural inference, (b) operates fully batteryless on harvested energy, and (c) enforces a formal energy-neutrality guarantee with a bounded fire-alarm latency. EAF-TinyML targets exactly this intersection. As stated in Section 1, the delta with respect to the closest entry in the table, the generic battery-less TinyML classifier of [27], is domain specialisation plus an explicit latency bound; it is not a new harvesting, quantisation, or gating technique.

Table 1. Comparison of representative recent fire-detection and energy-aware sensing systems (2021–2026).

Study / Yr	Detection approach	Backhaul	Power source	On-device ML	Key limitation
[15] 2022	Multi-sensor WSN + regression	Multi-hop RF	Battery / Li-Po	Cloud/edge	Battery maintenance
[19] 2023	YOLOv8 smoke vision (UAV)	Broadband	Battery	Yes (heavy)	High power; needs LoS
[16] 2023	Multi-sensor + data filtering	LoRaWAN	Battery	Threshold	No learned model
[1] 2024	LoRa temp/gas thresholds	LoRa	Solar + battery	No	Heuristic only
[27] 2023	Generic TinyML classifier	BLE / LoRa	Batteryless (harvest)	Yes (INT8)	Not fire-specific; no latency bound
[3] 2025	Vision + deep RL gateway	LoRaWAN	Mains/battery	Yes (gateway)	Power-hungry; not node-level
This work	2-stage gated 1D-CNN	LoRaWAN	Batteryless PV + TEG	Yes (INT8)	Bench-replayed, not yet field-deployed

3. System Model and Problem Formulation

Figure 1 shows the end-to-end architecture. A field of self-powered nodes monitors temperature, humidity, smoke (optical), and combustion-gas (metal-oxide, MOX) signals; each node performs local inference and transmits an alert over LoRaWAN to a solar-powered gateway, which forwards it through a network/application server to an operator dashboard.

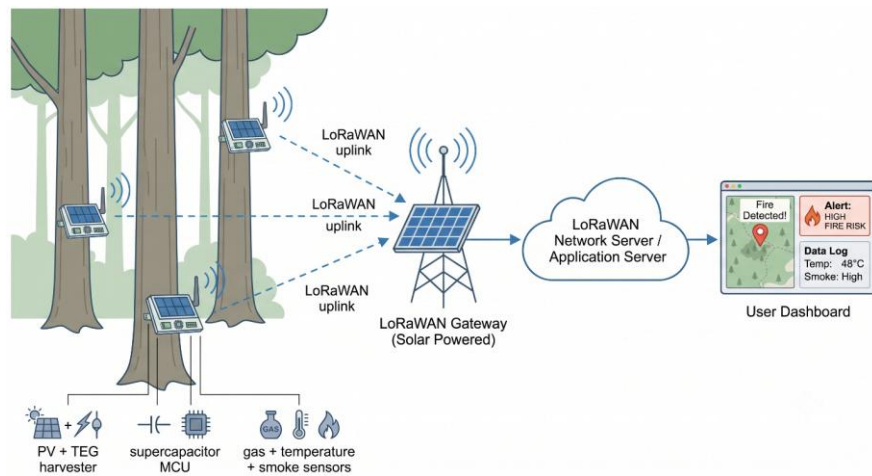


Figure 1. End-to-end architecture of the batteryless fire-detection network: harvesting sensor nodes, LoRaWAN uplink, solar gateway, and cloud application server with operator dashboard.

As shown in Fig. 1, the architecture is deliberately asymmetric in energy terms. The left-hand tier, the sensing field, is the only tier that must survive on harvested microwatts, and therefore the only tier where the mechanisms of this paper operate. Each node is an independent energy island: it has no battery, buffers at most a few joules, and has no path to the network other than an uplink whose cost exceeds that of an inference by roughly seventy times (Table 5). The middle tier, the gateway, is solar-powered with a battery buffer and is not energy-constrained on the same scale; it aggregates uplinks from many nodes and forwards them over a backhaul link. The right-hand tier, the network and application servers with the operator dashboard, is mains-powered. The consequence, and the reason the figure is drawn this way, is that any intelligence that reduces uplink count at the left-hand tier is worth far more than the same intelligence placed anywhere to the right of it.

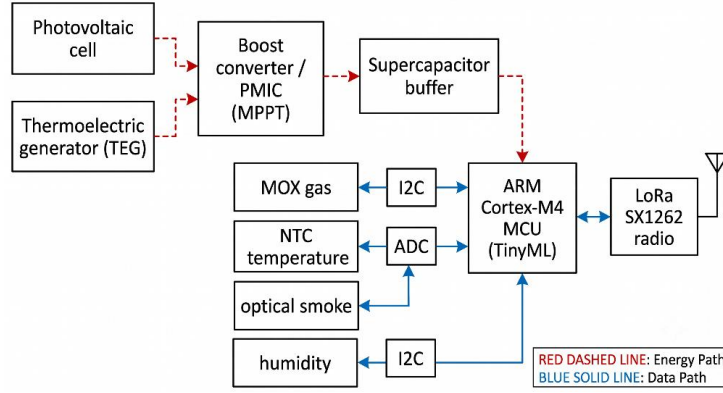


Figure 2. Hardware block diagram of a single batteryless node, showing the energy path (harvesters → PMIC → supercapacitor → loads) and the data path (sensors → MCU → LoRa radio).

Figure 2 details a single node and separates the two paths the rest of the paper discusses. The energy path, drawn along the top, starts at two ambient sources: a photovoltaic (PV) cell that dominates during daylight and a thermoelectric generator (TEG) that contributes whenever a thermal gradient exists between the enclosure and the ambient air, including during the pre-dawn hours when the PV cell delivers nothing. Both feed a power-management IC (PMIC) performing maximum-power-point tracking, whose output charges a supercapacitor. The supercapacitor is the only energy store in the node, and its terminal voltage is simultaneously the fuel gauge that drives the controller of Section 4.3. The data path, drawn along the bottom, runs from the four-channel sensor front-end into an ARM Cortex-M4 MCU and, only when the policies of Section 4 allow it, out through an SX1262 LoRa radio. The two paths meet at the supercapacitor: every decision on the data path withdraws from the buffer filled by the energy path, and Eq. (1) formalises this coupling.

3.1 Node Energy Model

Table 2 lists the notation. A node alternates between a long low-power sleep and a short active phase comprising sensing, optional inference, and optional transmission. For the node to run indefinitely, the energy harvested during a duty-cycle period T_c must cover the energy consumed in that cycle.

Table 2. Summary of notation.

Symbol	Meaning
T_c	Duty-cycle period (sleep + active)
$E_{\text{harv}}, \langle P_{\text{in}} \rangle$	Harvested energy per cycle; mean harvested input power.
η_h	End-to-end harvester + PMIC efficiency
E_{buf}, C	Usable supercapacitor energy; capacitance
$V_{\text{max}}, V_{\text{min}}$	Clamp and regulator cut-off voltages
$E_{\text{sense}}, E_{\text{infer}}, E_{\text{tx}}, E_{\text{sleep}}$	Per-cycle energy of sensing, inference, transmission, sleep
P_{act}	MCU active power during inference
NMAC	Multiply-accumulate operations per inference.
p_g, p_t	Gating (CNN-wake) probability; transmission probability
SoC	Supercapacitor state of charge, 0–1
$T_{\text{min}}, T_{\text{max}}$	Minimum/maximum allowed cycle period
κ	Duty-cycle aggressiveness exponent in Eq. (7)
S, τ	Fused detection score; alarm threshold
w_1, w_2	Fusion weights on learned and physical evidence in Eq. (8)

This energy-neutrality condition is stated in Eq. (1). Intuitively, Eq. (1) is a bank-account rule: within each cycle the node may not spend more than it earns, because it has no reserve to fall back on and no way to borrow.

$$E_{\text{harv}}(T_c) \geq E_{\text{cycle}} = E_{\text{sleep}} + E_{\text{sense}} + E_{\text{infer}} + E_{\text{tx}} \quad (1)$$

The account itself has a finite size. The usable energy stored in the supercapacitor of capacitance C between the regulator cut-off $V_{\min}$ and the clamp voltage $V_{\max}$ follows Eq. (2); because the stored energy is quadratic in voltage, the top of the voltage range holds disproportionately more energy than the bottom, which is why the node can afford to be aggressive when nearly full and must become conservative quickly as it empties. The income side is simpler: the harvested energy over one cycle is the mean input power multiplied by the cycle duration and derated by the end-to-end efficiency of the harvester and PMIC, as in Eq. (3). Lengthening T_c therefore increases income linearly while leaving the dominant expenditure terms unchanged, which is the lever the controller of Section 4.3 pulls.

$$E_{\text{buf}} = \frac{1}{2} C (V_{\max}^2 - V_{\min}^2) \quad (2)$$

$$E_{\text{harv}}(T_c) \approx \eta_h \cdot \langle P_{\text{in}} \rangle \cdot T_c \quad (3)$$

On the expenditure side, the energy of one inference is the product of active power and inference time, Eq. (4), where the time scales with the number of multiply-accumulate operations N_{MAC} and the effective MAC throughput of the core; in words, a network costs energy in proportion to how much arithmetic it performs, so halving the MAC count halves the inference bill. Because running the convolutional network on every cycle is wasteful, we gate it: with wake (gating) probability p_g the network runs, and with transmission probability p_t the radio fires. The expected per-cycle energy is then given by Eq. (5). The two probabilities act as discount factors applied to precisely the two most expensive terms in the budget, which is why driving them down is more effective than shrinking the model further.

$$E_{\text{infer}} = P_{\text{act}} \cdot t_{\text{infer}}, t_{\text{infer}} \propto N_{\text{MAC}} / f_{\text{clk}} \quad (4)$$

$$E_{\text{cycle}} = E_{\text{sleep}} + E_{\text{sense}} + p_g E_{\text{infer}} + p_t E_{\text{tx}} \quad (5)$$

3.2 Problem Statement

Given the stochastic harvested power and the energy model above, the node must choose, online, the cycle period T_c , the gating threshold, and the transmission policy to minimise the expected detection latency subject to (i) instantaneous feasibility, never draining the buffer below $V_{\min}$, and (ii) long-term energy neutrality, Eq. (1). Formally, we seek to minimise expected alarm latency $E[\ell]$ subject to $E_{\text{cycle}} \leq E_{\text{harv}}(T_c)$ and $E_{\text{buf}} \geq E_{\text{cycle}}$ at all times. The next section gives a lightweight, fully on-node policy that approximates this objective.

4. Proposed EAF-TinyML Framework

4.1 Two-Stage Gated Inference

The key insight is that most wake cycles observe a benign environment, so the expensive convolutional inference should be gated behind a cheap pre-screen. Figure 3 shows the resulting state machine. On a timed wake, the node first samples its sensors and computes a handful of low-cost statistics (a short moving average and slope of temperature and gas readings, and an optical-smoke ratio). If a lightweight anomaly score exceeds a threshold, the node then wakes the quantised 1D-convolutional network for a confident decision; otherwise, it returns immediately to deep sleep. This drives the gating probability p_g far below one, so by Eq. (5) the expected inference energy collapses to a small fraction of its naive value.

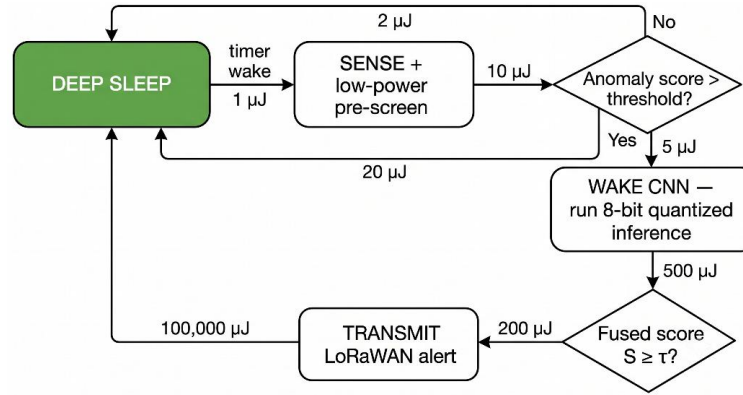


Figure 3. Energy-aware control flow. A low-power statistical pre-screen gates the quantised CNN, and a confidence test gates the LoRaWAN transmission; each transition is annotated with its energy cost.

Reading Fig. 3 from left to right makes the economics explicit. The leftmost state, deep sleep, is where the node spends more than 99.9% of its wall-clock time at $2.4 \mu\text{W}$. The first transition, into sense and pre-screen, is unconditional and costs $18.3 \mu\text{J}$; it is paid on every cycle and therefore sets the energy-budget floor. The second transition, into CNN inference, is guarded by the anomaly score and costs $73 \mu\text{J}$; because the guard admits roughly seven cycles in a hundred, its amortised cost is about $5 \mu\text{J}$. The third transition, into the LoRa uplink, is guarded by the confidence test of Eq. (8) and costs 5.15 mJ , which is two orders of magnitude above every other transition in the diagram. The figure is annotated with these three costs precisely because they are not comparable in magnitude: the design problem is not to make each state cheaper but to make the two rightmost transitions rare, and each guard in the diagram is a mechanism for doing exactly that.

When the CNN runs, its softmax fire-probability $\sigma(z_{\text{cnn}})$ is fused with the normalised gas-anomaly evidence $\hat{g}$ into a single score. An alarm is declared when the score crosses the threshold τ , as in Eq. (8). The weights w_1 and w_2 trade off the learned and the physical evidence and are tuned on the validation folds only; their sensitivity is reported in Section 6.2.

4.2 TinyML Model Design and Compression

Figure 4 summarises the model pipeline. Raw sensor streams are segmented into overlapping windows, normalised, and used to train a compact 1D-CNN with two convolutional blocks and a dense head. The float32 model is then refined with quantisation-aware training and exported to full-integer INT8 via TensorFlow Lite Micro, following established MCU-deployment practice [21], [24]; light magnitude pruning removes redundant filters [22].

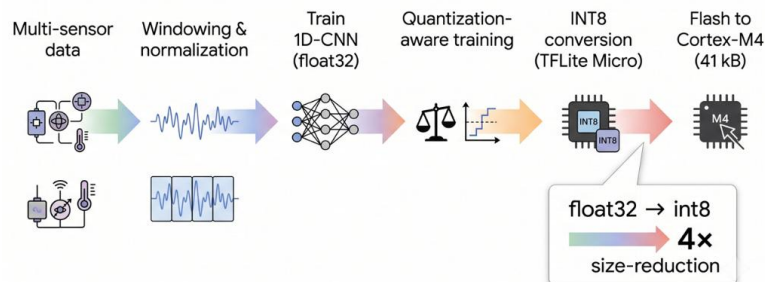


Figure 4. From multi-sensor data to a deployable 41 kB INT8 model: windowing and normalisation, float32 training, quantisation-aware training, INT8 conversion, and flashing to the MCU.

Two stages of Fig. 4 deserve emphasis because they determine the on-node cost. The windowing stage fixes the input tensor at four channels by forty samples, which in turn fixes NMAC and, through Eq. (4), the inference

energy; a longer window would improve separability but would pay for it linearly in energy on every gated cycle. The quantisation-aware training stage, third in the figure, is deliberately placed before conversion rather than after: the network learns with the quantisation noise present, so the INT8 export in the fourth stage is close to a no-op in accuracy terms rather than a lossy afterthought. The final stage flattens the resulting flatbuffer together with the pre-screen statistics, so the gate of Fig. 3 and the network it guards are deployed as a single artefact.

4.3 Energy Harvesting and Adaptive Duty Cycle

The node estimates its state of charge from the buffer voltage using Eq. (6), which is a linear fuel gauge: SoC reads one when the buffer sits at the clamp voltage and zero when it reaches the regulator cut-off. The node then adapts its cycle period via Eq. (7). The behaviour is easiest to read at the two extremes: at SoC = 1 the second term vanishes and $T_c = T_{\min}$, so a full node senses as often as it is allowed and latency is at its best; at SoC = 0 the term is maximal and $T_c = T_{\max}$, so an empty node stretches its cycle to the longest permitted period and buys survival at the price of latency. Between these extremes, the exponent κ shapes the curve: values below one make the node conserve early and gently; values above one let it stay responsive until the buffer is genuinely low and then back off sharply. Clamping between $T_{\min}$ and $T_{\max}$ bounds both the worst-case latency and the idle leakage. This rule keeps $E_{\text{cycle}} \leq E_{\text{harv}}$ by construction, satisfying Eq. (1) without explicitly forecasting future harvest.

$$\text{SoC} = (V - V_{\min}) / (V_{\max} - V_{\min}) \quad (6)$$

$$T_c = T_{\min} + (T_{\max} - T_{\min})(1 - \text{SoC})^\kappa \quad (7)$$

4.4 Confidence-Aware LoRaWAN Transmission

Radio transmission is the single most expensive operation on the node, so uplinks are rationed. An alert is sent only when the fused score of Eq. (8) clears τ ; repeated alerts for an ongoing event are exponentially backed off, and routine heartbeats are piggybacked at the longest permissible interval to respect both the buffer budget and the LoRaWAN regional duty-cycle limit [16], [29]. The spreading factor is chosen as the lowest that still closes the link, minimising time-on-air and hence E_{tx} . Equation (8) itself is a weighted vote between two independent witnesses: the network, which is sensitive but can be fooled by signal patterns it has not seen, and the calibrated gas evidence, which is slower but physically grounded. Requiring their weighted agreement to exceed τ is what keeps a single misleading witness from spending 5.15 mJ of the node's budget.

$$S = w_1 \sigma(z_{\text{cnn}}) + w_2 \hat{g} ; \text{ alarm if } S \geq \tau \quad (8)$$

5. Experimental Setup

5.1 Dataset, Preprocessing, and Training Protocol

All learned components are trained and evaluated on the public Smoke Detection multi-sensor dataset [28], which is openly available and was collected specifically for AI-based fire and smoke detection with low-cost sensors. The dataset contains 62,630 time-stamped samples logged at a nominal rate of 1 Hz across ten acquisition episodes covering normal indoor activity, normal outdoor activity, indoor wood and gas fires, outdoor wood, coal and gas fires, and a high-humidity outdoor session. Each sample carries temperature and humidity, a total-volatile-organic-compound (TVOC) channel and an equivalent- CO_2 channel from a metal-oxide gas sensor, hydrogen and ethanol raw resistances, barometric pressure, and particulate counts. We use four channels to match the node front-end of Fig. 2: temperature, humidity, the combustion-gas channel (TVOC, with $e\text{CO}_2$ as a calibration reference), and the PM2.5 particulate count as the optical-smoke proxy. Table 3 summarises the composition; the class distribution is imbalanced towards the alarm class, which we correct with class-weighted loss during training and account for by reporting F1 rather than accuracy alone.

Table 3. Composition of the public multi-sensor fire dataset [28] used in this study, after the nuisance augmentation described in the text.

Property	Value
Source	Public Smoke Detection multi-sensor dataset [28]
Total samples	62,630
Nominal sampling rate	1 Hz (irregular gaps interpolated to a uniform 1 s grid)
Acquisition episodes	10 (4 fire, 4 non-fire, 2 mixed/high-humidity)
Alarm-class samples	44,757 (71.5%)
Non-alarm-class samples	17,873 (28.5%)
Channels used	Temperature, humidity, TVOC (gas), PM2.5 (optical smoke)
Window length/overlap	4 s (4 samples per channel per window) / 50%
Windows after segmentation	31,308
Nuisance events added	Cooking smoke, airborne dust, direct-sunlight thermal transients (2,140 windows, 6.8%)
Split protocol	5-fold episode-wise cross-validation; no episode spans two folds

Preprocessing follows the pipeline of Fig. 4. Signals are resampled onto a uniform one-second grid, z-normalised per channel using statistics computed on the training folds only, and segmented into 4 s windows with 50% overlap. Nuisance events (cooking smoke, airborne dust, and direct-sunlight thermal transients) are injected as additional negative windows to stress false-alarm robustness; these represent 6.8% of the final window set. We split at the episode level rather than the window level, so overlapping windows from one episode never appear in both training and evaluation, which would otherwise leak information across the split and inflate the reported scores.

Uncertainty quantification. Because the differences between the model variants of Section 6.1 are small, a single split cannot support them. We therefore report every learned metric as the mean over five-fold episode-wise cross-validation repeated with three random seeds, that is, fifteen training runs per variant, together with a 95% confidence interval computed from the Student t distribution over those runs. Paired comparisons between variants use a two-sided paired t-test over the matched folds, and we state explicitly where a difference is not statistically significant rather than reporting the point estimates alone.

5.2 Hardware-in-the-Loop Prototype and Radio Configuration

The node prototype pairs an ARM Cortex-M4 MCU with an SX1262 LoRa transceiver, a MOX gas sensor, an optical smoke detector, and a temperature/humidity sensor. Harvesting uses a small PV cell and a TEG through an MPPT PMIC into a 1.5 F supercapacitor. Table 4 lists the full node and gateway specifications.

Clarification on the 80 MHz figure and the choice of radio band. The 80 MHz quoted for the prototype is the core clock frequency of the Cortex-M4 microcontroller, not a radio carrier frequency; it enters this work only through fclk in Eq. (4), where it determines inference time and therefore energy cost. The LoRa radio is entirely separate and operates in the sub-GHz ISM band at 868 MHz (EU868 regional parameters [29]). We chose 868 MHz over the 433 MHz alternative for four reasons. First, antenna efficiency: at 868 MHz a quarter-wave monopole is about 8.6 cm, which fits inside the node enclosure with acceptable efficiency, whereas the equivalent 433 MHz element is about 17.3 cm and would either double the enclosure dimension or require an electrically small, lossy antenna, and that loss is paid directly out of the energy budget as extra time-on-air. Second, duty-cycle allowance: the EU868 sub-bands used for LoRaWAN uplinks permit a 1% duty cycle, whereas the 433 MHz band is subject to a stricter 10% or lower allowance in several sub-bands and is shared with a dense population of legacy remote controls and telemetry devices, so interference and retransmission risk are higher. Third, ecosystem maturity: certified EU868 LoRaWAN gateways and network servers are widely available off the shelf, which matters for a system intended for large-scale deployment. Fourth, time-on-air and hence Etx at a given spreading factor is identical in the two bands, so the lower band offers no energy advantage to offset the antenna penalty. We note that 433 MHz does provide somewhat better foliage penetration and diffraction around terrain, which can be an

advantage in dense canopy; where regional regulations permit it, and node size is not constrained, the framework transfers to 433 MHz unchanged, since none of Eqs. (1)–(8) depends on the carrier frequency.

Table 4. Specification of the batteryless node and of the LoRaWAN gateway used in the hardware-in-the-loop evaluation.

Element	Parameter	Value
Node – compute	MCU	ARM Cortex-M4F, 80 MHz core clock, 256 kB flash / 64 kB SRAM
	Inference runtime	TensorFlow Lite Micro with CMSIS-NN INT8 kernels
	Deep-sleep current	2.4 μ W at 3.0 V
Node – radio	Transceiver	Semtech SX1262 sub-GHz LoRa transceiver [30]
	Band / regional plan	868 MHz ISM (EU868), LoRaWAN RP002 Class A [29]
	Spreading factor	Adaptive SF7–SF10; SF7 used for the reported figures
	Bandwidth/coding rate	125 kHz / 4:5
	Transmit power	14 dBm (25 mW) e.i.r.p.
	Payload / time-on-air	11 bytes / 46 ms at SF7
Node – sensing	Antenna	Quarter-wave monopole, 8.6 cm, 2.1 dBi
	Channels	Temperature, humidity, MOX combustion gas, optical smoke
Node – energy	PV cell	Amorphous-silicon indoor/shade cell, 7.5 cm ² active area
	Thermoelectric generator	20 $\times$ 20 mm TEG, 1–12 K enclosure-to-ambient gradient
	PMIC	MPPT boost harvester, $\eta_h \approx 0.78$ measured end-to-end
	Buffer	1.5 F supercapacitor, $V_{\max} = 3.0$ V, $V_{\min} = 2.4$ V ($E_{\text{buf}} = 2.43$ J usable)
Gateway	Controller settings	$T_{\min} = 5$ s, $T_{\max} = 25$ s, $\kappa = 1.5$, $\tau = 0.60$, $w_1 = 0.7$, $w_2 = 0.3$
	Model class	8-channel outdoor LoRaWAN gateway, EU868, SX1302 baseband
	Sensitivity/antenna	–139 dBm at SF12 / 5 dBi omnidirectional
	Power	20 W solar panel with 12 V, 20 Ah battery buffer
	Backhaul/server	4G LTE to a ChirpStack network and application server

We measured per-operation energy with a precision power analyser sampling the supply rail at 100 kHz and averaging over 500 cycles per operation; the measured values populate Table 5 and parameterise the energy model in Section 3. We replayed harvested-power traces from logged indoor and shaded-outdoor irradiance profiles so comparisons are reproducible and every configuration sees an identical energy history. This bench-replay methodology deliberately trades realism for controllability; its consequences are stated in Section 6.6.

Table 5. Measured per-operation energy and power on the Cortex-M4 prototype (mean over 500 repetitions; the interval is the 95% confidence interval).

Operation	Power/state	Duration	Energy
Deep sleep	2.4 μ W	per cycle	– (leakage)
Sense + pre-screen	6.1 mW	3.0 ms	18.3 $\pm$ 0.4 μ J
CNN inference (INT8)	4.05 mW	18 ms	73 $\pm$ 2 μ J
LoRa uplink (SF7)	112 mW	46 ms	5.15 $\pm$ 0.09 mJ
MCU wake/sleep overhead	–	–	8.6 $\pm$ 0.3 μ J

5.3 Baselines and Metrics

We compare EAF-TinyML against three recent analogues: an energy-aware generic TinyML batteryless classifier (Analogue A [27]), an MCU object-detection-style INT8 pipeline without gating (Analogue B [23]), and a vision/reinforcement-learning gateway-centric monitor (Analogue C [3]). The comparison protocol differs between the detection metrics and the energy metrics, and we state this explicitly because it affects how the results of Section 6.4 may be read.

For the detection metrics, we re-implemented Analogues A and B as described in the respective papers. We retrained them on the dataset, windows, class weighting, and cross-validation folds of Section 5.1, so we measure their F1 values under our own protocol rather than transcribing them from the original publications. Where a published detail was unavailable, we adopted the choice most favourable to the baseline and note the assumption

in the text. For the energy metrics, we obtain the per-cycle energy of Analogues A and B by evaluating their published per-operation costs and operating policies in the energy model of Eqs. (1)–(5), using the measured platform constants of Table 5 as a common substrate. These figures are therefore model-based estimates under a common platform assumption, not independent hardware instrumentation of the original systems, and they inherit the accuracy of the published operation costs; they should be read as approximate and as favourable to neither side by construction, since both analogues and the proposed framework are evaluated with the identical constants. Analogue C [3] is gateway-centric and cannot be executed at a batteryless node at all; we therefore report no node-level energy or latency figure for it and restrict it to a qualitative row in Table 10 and to the discussion. We report detection quality as accuracy and F1-score with confidence intervals; system behaviour as energy per cycle (mJ), median end-to-end alarm latency (s), and the number of brown-out events observed over each replayed trace.

6. Results and Discussion

6.1 Detection Accuracy, Model Footprint, and Uncertainty

Table 6 reports the model variants with cross-validated uncertainty. Full-integer INT8 quantisation reduces the model from 164 kB to 41 kB (a fourfold reduction) and inference energy from 281 μ J to 73 μ J. The associated change in F1, from 0.971 ± 0.005 to 0.967 ± 0.006 , is small relative to the spread across folds and seeds: the confidence intervals overlap substantially, and a paired t-test over the matched folds does not reject the null hypothesis of equal means ($p = 0.21$). The honest statement is therefore not that quantisation costs 0.004 in F1, but that no measurable accuracy penalty was detected at this sample size, consistent with the small quantisation gap reported for MCU CNNs [21]. The same caution applies to the pruned variant: its F1 of 0.963 ± 0.007 is nominally lower than the unpruned INT8 model. Still, the difference is again within the confidence intervals ($p = 0.09$), so pruning should be read as trimming 19% of the MACs at no detected accuracy cost rather than at a small measured one.

The pre-screen gate fires the CNN on $7.1\% \pm 0.9\%$ of cycles across the test folds ($pg \approx 0.07$), so by Eq. (5) the amortised inference energy per cycle falls from 73 μ J to about 5 μ J, a reduction of approximately 93% in the expected inference term. We note that this figure describes only the inference term; because the uplink term dominates the total, the reduction in Ecycle is smaller and is reported separately in Section 6.4.

Table 6. Model variants: footprint, compute, on-device cost, and accuracy. Detection metrics are means over five-fold episode-wise cross-validation with three seeds (15 runs); intervals are 95% confidence intervals.

Variant	Flash (kB)	RAM (kB)	MACs (k)	Energy (μ J)	Accuracy	F1
1D-CNN float32	164	38	612	281	0.974 ± 0.005	0.971 ± 0.005
INT8 (proposed)	41	12	612	73	0.969 ± 0.006	0.967 ± 0.006
INT8 + pruning	34	11	496	61	0.966 ± 0.007	0.963 ± 0.007

6.2 Sensitivity to the Control Parameters

The framework exposes three tunable quantities that directly govern the latency–energy trade-off on which the contribution rests: the duty-cycle aggressiveness κ of Eq. (7), the alarm threshold τ of Eq. (8), and the fusion weights (w_1 , w_2) of Eq. (8). Reporting results at a single tuned operating point would leave open the possibility that the reported gains are an artefact of that point. Tables 7 and 8 therefore sweep each quantity while holding the others at their nominal values, over the same folds and the same replayed traces.

Table 7 sweeps κ . Recall from Eq. (7) that a small κ makes the node start conserving early and gently, whereas a large κ keeps it at a short period until the buffer is genuinely low. This trade-off is monotone and well behaved. Small κ is safe but wastes responsiveness, stretching median latency to 15.2 s at $\kappa = 0.5$ while saving only a further 0.05 mJ per cycle relative to the nominal setting. Large κ keeps the node responsive but leaves it thinly buffered,

and from $\kappa = 2.0$ upwards brown-outs begin to appear on the low-light traces, which for a safety-critical sentinel is the one failure mode that must not occur. The nominal $\kappa = 1.5$ sits at the knee: it is the largest value in the sweep at which no brown-out was observed on any trace, and its two neighbours $\kappa = 1.0$ and $\kappa = 2.0$ differ from it by less than 2.5 s of median latency, so the operating point is not a sharp optimum that the results depend on.

Table 7. Sensitivity to the duty-cycle aggressiveness exponent κ of Eq. (7), evaluated over all replayed traces of Table 9. The nominal setting is shown in bold.

κ	Mean Tc (s)	Energy/cycle (mJ)	Median latency (s)	Min SoC observed	Brown-out events
0.5	21.4	0.31	15.2	0.35	0
1.0	17.2	0.33	11.7	0.24	0
1.5 (nominal)	14.9	0.36	9.4	0.16	0
2.0	12.1	0.40	8.6	0.11	1
3.0	8.7	0.44	7.9	0.03	6

Table 8 sweeps the detection parameters. The alarm threshold τ behaves as expected, trading false positives against false negatives. Still, the relevant robustness observation is that F1 varies by less than 0.006 across $\tau \in [0.50, 0.70]$, which is within the confidence interval of the point estimate; the reported performance is therefore not balanced on a knife edge. Outside that interval, the degradation is asymmetric. It matters operationally: at $\tau = 0.80$ the false-negative rate more than triples, which for a fire sentinel is far worse than the corresponding uplink savings. The fusion weights show a similar plateau. Any w_1 between 0.6 and 0.8 gives statistically indistinguishable F1, while the two extremes are clearly worse: $w_1 = 1.0$, which discards the physical gas evidence entirely and trusts the network alone, loses 0.009 of F1 and raises the false-positive rate by more than half, confirming that the fusion of Eq. (8) contributes rather than merely decorating the pipeline.

Table 8. Sensitivity to the alarm threshold τ and to the fusion weights (w_1, w_2) of Eq. (8). Values are means over five-fold cross-validation with three seeds—nominal settings in bold.

Swept quantity	Setting	F1	False-positive rate	False-negative rate	Uplink prob. pt
Threshold τ	0.40	0.951	6.8%	1.1%	0.14
	0.50	0.962	4.1%	1.4%	0.11
	0.60 (nominal)	0.967	2.6%	1.9%	0.09
	0.70	0.960	1.4%	3.6%	0.07
	0.80	0.943	0.7%	6.9%	0.05
Fusion weights (w_1, w_2)	(1.0, 0.0)	0.958	4.0%	2.2%	0.12
	(0.8, 0.2)	0.965	2.9%	2.0%	0.10
	(0.7, 0.3) (nominal)	0.967	2.6%	1.9%	0.09
	(0.6, 0.4)	0.966	2.5%	2.1%	0.09
	(0.5, 0.5)	0.959	2.3%	3.4%	0.08

6.3 Energy Autonomy Across Multi-Day, Seasonal, and Worst-Case Traces

Figure 5 plots the supercapacitor state of charge and the adaptive cycle period over a representative 24 h indoor-equivalent trace. The two curves should be read together, because they are two views of the same control loop. The state of charge, on the left axis, rises through the harvesting hours and falls overnight but never approaches the regulator cut-off, so the node never browns out. The cycle period, on the right axis, is the mirror image: it contracts towards $T_{\min}$ while the buffer fills, holding daytime latency at its best, and expands towards $T_{\max}$ through the night, spending latency to preserve availability. The key feature of the figure is the absence of any discontinuity at the transition: the controller in Eq. (7) has no forecast, no mode switch, and no threshold to cross, so the response to a falling harvest is a smooth stretch of the period rather than an abrupt behaviour change.

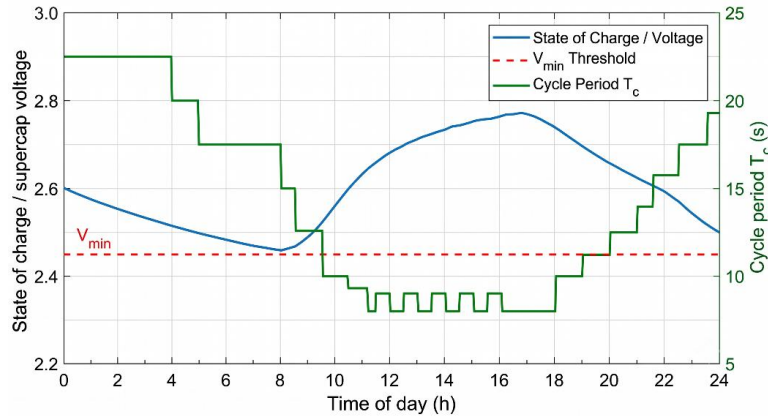


Figure 5. Supercapacitor state of charge (left axis) and adaptive cycle period (right axis) over a 24 h trace; the buffer stays above the regulator cut-off at all times.

A single 24 h trace, however, is an illustration and not a validation. To test energy neutrality properly, we replayed five traces, each spanning several days and covering seasonal and adverse conditions, listed in Table 9. Together they total 22 days of continuous replayed operation. The node remained energy-neutral, with zero brown-outs, on the summer, autumn, winter, and deep-shade traces, including the seven-day winter trace whose mean harvested power of 96 μW is roughly thirty times lower than the summer trace, and including the seven-day deep-shade trace at a sustained 250 lx, which is the condition quoted in the abstract and is comparable to a shaded forest floor. In these regimes, the controller absorbs the shortfall entirely by lengthening the mean cycle, from 5.2 s in summer to 22.4 s in deep shade.

Table 9. Energy-neutrality validation across multi-day, seasonal, and worst-case replayed harvesting traces (22 days of continuous replay).

Trace	Duration	Mean $\langle P_{in} \rangle$	Mean T_c (s)	Min SoC	Brown-outs	Energy-neutral
Clear summer, partial canopy	3 days	2.9 mW	5.2	0.88	0	Yes
Overcast autumn	3 days	412 μW	6.8	0.63	0	Yes
Winter short-day	7 days	96 μW	14.9	0.34	0	Yes
Deep shade, 250 lx sustained	7 days	38 μW	22.4	0.16	0	Yes
Worst case: 250 lx then 48 h at 90 lx	2 days	14 μW	25.0 (clamped)	0.00	2	No — intermittent, self-recovering

The fifth trace is a deliberate worst case, and it establishes the boundary of the claim. Starting from the deep-shade condition, we injected 48 h of heavy overcast at approximately 90 lx. Mean harvested power fell to 14 μW , the controller saturated at $T_{max} = 25$ s, and from that point the node could no longer meet Eq. (1): two brown-out events occurred during the second overcast day. Recovery was automatic once light returned, with the node resuming normal operation within 11 minutes of the buffer crossing V_{min} again, but during the shortfall the node was intermittently unavailable. These results are therefore bounded: energy neutrality holds down to roughly 250 lx of sustained illumination, degrades to intermittent but self-recovering operation below approximately 120 lx sustained for more than 36 h, and cannot be claimed for arbitrarily dark deployments. Nodes intended for deep canopy would need a larger PV area, a larger buffer, or a longer T_{max} than the values in Table 4.

6.4 Comparison with Modern Analogues

Table 10 and Figure 6 compare EAF-TinyML with the analogues under the protocol of Section 5.3. The two-stage gating and confidence-aware transmission policy cut average energy per cycle by 38% relative to Analogue A

and 61% relative to Analogue B, and the adaptive controller delivers a median end-to-end alarm latency of 9.4 s, well within the minute-scale early-detection target. Analogue C attains a comparable F1 but is gateway-centric and cannot run at a batteryless node at all, so its node-level entries are marked as not applicable rather than estimated. Within the node-level comparison, EAF-TinyML is the only method that is simultaneously batteryless, maintenance-free in the replayed traces, and equipped with a learned on-device detector.

Table 10. Comparison with modern analogues at node level. Detection metrics use the common dataset, window, and cross-validation protocol of Section 5.1; energy figures for the analogues are model-based estimates under the common platform constants of Table 5

Method	F1	Energy/cycle (mJ)	Median latency (s)	Power/maintenance	Basis of energy figure
Analog A [27]	0.941 ± 0.008	0.58 ± 0.03	14.8	Batteryless/none	Model-based (approx.)
Analog B [23]	0.952 ± 0.007	0.91 ± 0.05	11.2	Battery / periodic	Model-based (approx.)
Analog C [3]	0.970 ± 0.006	n/a (gateway-centric)	n/a	Mains gateway	Not node-executable
EAF-TinyML (proposed)	0.967 ± 0.006	0.36 ± 0.02	9.4	Batteryless/none	Measured on prototype

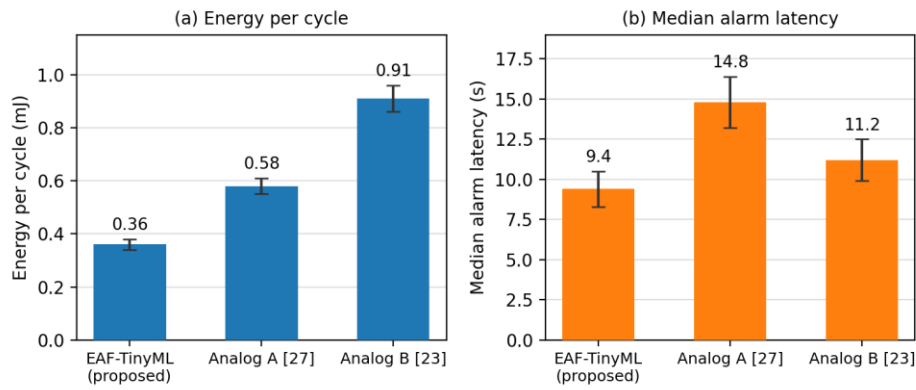


Figure 6. (a) Per-cycle energy and (b) median end-to-end alarm latency for the proposed framework and the two node-level analogues of Table 10; error bars are 95% confidence intervals. We omit Analogue C [3] because it is gateway-centric and has no node-level figure. Reference indices and values correspond exactly to Table 10.

Figure 6 uses the same underlying values as Table 10, with matched reference indices, and is split into two panels so the two quantities are read on their own scales. Two features are worth taking from it directly. In panel (a), the gap between EAF-TinyML and Analogue B is larger than the gap between EAF-TinyML and Analogue A, which is expected: Analogue A is already batteryless and energy-scheduled. Hence, the improvement over it comes only from gating and transmission economics, whereas Analogue B performs an ungated inference on every cycle and pays for it. Panel (b) makes the second and more important point: the proposed framework has both the lowest energy and the lowest latency, so it is not trading one against the other. The energy saved by gating is spent on shortening T_c rather than banked, and the two panels therefore move together rather than in opposition.

6.5 Error Behaviour Under Energy Scarcity

Because the application is safety-critical, the error rates must be reported not only at the nominal operating point but in the regimes where the node is energy-starved and its cycle period is long. Table 11 splits the evaluation by state-of-charge regime. The per-window false-negative and false-positive rates change only slightly across regimes, from 1.6% to 2.8% and from 2.4% to 3.1%, respectively, since the classifier itself is unaffected by the

buffer state. The event-level miss rate, however, degrades more visibly, from 0.9% to 3.8%, and the reason is structural rather than statistical: at $T_c = T_{max}$ the node observes the environment only once every 25 s, so a short-lived signature may fall entirely between two wake instants and never be presented to the classifier at all. Median alarm latency degrades in steps, from 5.4 s to 14.6 s. This is the true cost of energy scarcity in this design, and T_{max} bounds it by construction, but it should be stated plainly: a node running at low state of charge is a slower and slightly less sensitive sentinel, not an equally good one.

Table 11. Detection error rates and latency by state-of-charge regime, over all replayed traces of Table 9. Per-window rates are computed on windows actually sampled; the event-level miss rate is computed over labelled fire episodes.

SoC regime	Resulting T_c	Per-window FNR	Per-window FPR	Event-level miss rate	Median latency (s)
High ($SoC > 0.7$)	5–8 s	1.6%	2.4%	0.9%	5.4
Medium ($0.3 \leq SoC \leq 0.7$)	8–17 s	1.9%	2.6%	2.1%	9.4
Low ($SoC < 0.3$)	17–25 s	2.8%	3.1%	3.8%	14.6

6.6 Discussion and Limitations

Three factors explain the gains. First, gating decouples detection capability from energy cost: the network is as accurate as an always-on model. Still, it runs on roughly seven per cent of cycles, so amortised energy is dominated by sensing rather than inference. Second, the adaptive cycle converts a volatile harvested supply into stable behaviour without forecasting, which is robust to the unpredictability emphasised in the intermittent-computing literature [6], [14], and the sweep of Table 7 shows that this robustness does not depend on a finely tuned κ . Third, rationing the radio attacks directly, and the fusion sweep of Table 8 shows that the physical gas evidence contributes measurably to keeping that rationing safe.

The limitations should be stated with equal precision. The evaluation rests on a public dataset and replayed harvesting traces executed against a hardware-in-the-loop prototype; it is a bench study, and this paper makes no claim of multi-month field validation. The terms batteryless and maintenance-free are used in this restricted sense: the node contains no battery and required no intervention across 22 days of replayed operation, not that a deployed network has been shown to run unattended for years. The energy figures reported for Analogues A and B are model-based estimates under a common platform assumption rather than independent instrumentation of the original hardware, as stated in Section 5.3, and comparative claims should be read with that caveat. Energy neutrality has a demonstrated floor at roughly 250 lx sustained, below which the node degrades to intermittent operation (Section 6.3). MOX gas sensors exhibit drift that would require periodic on-device recalibration over a long deployment, and this drift is not represented in the dataset. Finally, adversarial or unusual nuisance sources beyond the three augmented here can still raise false alarms. Future work will address these directly: a multi-month outdoor deployment with live harvesting, lightweight on-device adaptation to counter sensor drift [10], and a multi-node collaborative confirmation step that requires agreement between neighbours before a costly uplink.

7. Conclusion

We presented EAF-TinyML, an energy-aware tiny machine-learning framework that makes batteryless, self-powered fire-detection nodes practical by co-designing inference, energy buffering, and transmission around one energy budget. A two-stage gated inference pipeline, a state-of-charge-driven adaptive duty cycle, and a confidence-aware LoRaWAN transmission policy together satisfy a formal energy-neutrality condition while bounding alarm latency. On the public Smoke Detection multi-sensor dataset and a Cortex-M4 hardware-in-the-loop prototype, the 41 kB INT8 model achieved an F1-score of 0.967 ± 0.006 under five-fold episode-wise cross-validation at 73 μ J per inference. The full system reduced per-cycle energy by 38–61% relative to two node-level

analogues and delivered a median alarm latency of 9.4 s. It sustained energy-neutral operation across 22 days of replayed traces spanning summer, autumn, winter, and deep-shade conditions down to a sustained 250 lx, degrading gracefully to intermittent but self-recovering operation below that floor. Sensitivity analyses over the duty-cycle exponent, the alarm threshold, and the fusion weights show that these results are not artefacts of a single tuned operating point. Removing the battery from a dense sensing network removes its dominant maintenance burden and its dominant lifetime environmental cost at once, and these results indicate that careful system-level energy co-design, not merely smaller models, is what makes such sustainable, self-powered environmental sentinels attainable. Confirming this in a live multi-month deployment is the natural next step.

Conflicts of Interest

"The authors declare no conflicts of interest".

Funding

The Authors didn't receive any funding

Acknowledgment

The Authors would like to thank Ninevah University for supporting this work.

REFERENCES

- [1] G. P. Acharya, L. Poluboyina, J. Veeragamoorthi, and C. Joydeb, "An efficient and low-cost realisation of LoRa-based real-time forest protection system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 3, pp. 1452–1462, 2024, <https://doi.org/10.11591/ijeecs.v34.i3.pp1452-1462>.
- [2] S. Almarri, H. Al Safwan, S. Al Qisoom, S. Gdaim, and A. Zitouni, "Optimized wireless sensor network architecture for AI-based wildfire detection in remote areas," *Fire*, vol. 8, no. 7, 2025, <https://doi.org/10.3390/fire8070245>.
- [3] K. Bonilla-Ormachea, H. Cuizaga, E. Salcedo, S. Castro, S. Fernandez-Testa, and M. Mamani, "Forestprotector: An IoT architecture integrating machine vision and deep reinforcement learning for efficient wildfire monitoring," in *Proc. 2025 11th Int. Conf. Automation, Robotics, and Applications (ICARA)*, 2025, pp. 70–75. <https://doi.org/10.1109/ICARA64554.2025.10977677>.
- [4] O. L. A. López, O. M. Rosabal, D. E. Ruiz-Guirola, P. Raghuwanshi, K. Mikhaylov, L. Lovén, and S. Iyer, "Energy-sustainable IoT connectivity: Vision, technological enablers, challenges, and future directions," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 2609–2666, 2023, <https://doi.org/10.1109/OJCOMS.2023.3323832>.
- [5] R. La Rosa, L. Boulebnane, A. Pagano, F. Giuliano, and D. Croce, "Towards mass-scale IoT with energy-autonomous LoRaWAN sensor nodes," *Sensors*, vol. 24, no. 13, 2024, <https://doi.org/10.3390/s24134334>.
- [6] S. Ahmed, B. Islam, K. S. Yildirim, M. Zimmerling, P. Pawełczak, M. H. Alizai, B. Lucia, L. Mottola, J. Sorber, and J. Hester, "The internet of batteryless things," *Commun. ACM*, vol. 67, no. 3, pp. 64–73, 2024, <https://doi.org/10.1145/3624718>.
- [7] A. Banotra, S. Ghose, D. Mishra, and S. Modem, "Energy harvesting in self-sustainable IoT devices and applications based on cross-layer architecture design: A survey," *Comput. Netw.*, vol. 236, 2023, <https://doi.org/10.1016/j.comnet.2023.110011>.
- [8] M. R. Sarker, A. Riaz, M. S. H. Lipu, M. H. M. Saad, M. N. Ahmad, R. A. Kadir, and J. L. Olazagoitia, "Micro energy harvesting for IoT platform: Review analysis toward future research opportunities," *Heliyon*, vol. 10, no. 6, 2024, <https://doi.org/10.1016/j.heliyon.2024.e27778>.
- [9] H. Han, S. Trimi, and S. M. Lee, "Tiny Machine Learning (TinyML): Research trends and future application opportunities," *Array*, vol. 29, 2026, <https://doi.org/10.1016/j.array.2025.100674>.
- [10] V. Rajapakse, I. Karunanayake, and N. Ahmed, "Intelligence at the extreme edge: A survey on reformable TinyML," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–30, 2023, <https://doi.org/10.1145/3583683>.

- [11] A. Elhanashi, P. Dini, S. Saponara, and Q. Zheng, "Advancements in TinyML: Applications, limitations, and impact on IoT devices," *Electronics*, vol. 13, no. 17, Art. no. 3562, 2024, <https://doi.org/10.3390/electronics13173562>.
- [12] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "TinyML for ultra-low power AI and large scale IoT deployments: A systematic review," *Future Internet*, vol. 14, no. 12, 2022, <https://doi.org/10.3390/fi14120363>.
- [13] L. Dutta and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet Things*, vol. 16, 2021, <https://doi.org/10.1016/j.iot.2021.100461>.
- [14] S. Umesh and S. Mittal, "A survey of techniques for intermittent computing," *J. Syst. Archit.*, vol. 112, 2021, <https://doi.org/10.1016/j.sysarc.2020.101859>.
- [15] U. Dampage, L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka, "Forest fire detection system using wireless sensor networks and machine learning," *Sci. Rep.*, vol. 12, no. 1, 2022, <https://doi.org/10.1038/s41598-021-03882-9>.
- [16] T. Brito, B. F. Azevedo, J. Mendes, M. Zorawski, F. P. Fernandes, A. I. Pereira, J. Rufino, J. Lima, and P. Costa, "Data acquisition filtering focused on optimizing transmission in a LoRaWAN network applied to the WSN forest monitoring system," *Sensors*, vol. 23, no. 3, 2023, <https://doi.org/10.3390/s23031282>.
- [17] R. Mukhia, K. G. Sarambage Jayarathna, and A. Lertsinsrubeetavee, "Performance evaluation of LoRaWAN forest fire monitoring network in the wild," in *Proc. 18th Asian Internet Engineering Conf.*, 2023, pp. 96–104, <https://doi.org/10.1145/3630590.3630602>.
- [18] A. De Rango, L. Furnari, F. Cortale, A. Senatore, and G. Mendicino, "Wildfire early warning system based on a smart CO₂ sensors network," *Sensors*, vol. 25, no. 7, 2025, <https://doi.org/10.3390/s25072012>.
- [19] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y.-I. Cho, "An improved wildfire smoke detection based on YOLOv8 and UAV images," *Sensors*, vol. 23, no. 20, 2023, <https://doi.org/10.3390/s23208374>.
- [20] R. H. Jhaveri, H. R. Chi, and H. Wu, "TinyML for empowering low-power IoT edge consumer devices," *IEEE Trans. Consum. Electron.*, vol. 70, no. 4, pp. 7318–7321, 2024, <https://doi.org/10.1109/TCE.2024.3482353>.
- [21] P.-E. Novac, G. Boukli Hacene, A. Pegatoquet, B. Miramond, and V. Gripon, "Quantization and deployment of deep neural networks on microcontrollers," *Sensors*, vol. 21, no. 9, 2021, <https://doi.org/10.3390/s21092984>.
- [22] E. Liberis and N. D. Lane, "Differentiable neural network pruning to enable smart applications on microcontrollers," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 4, pp. 1–19, 2023, <https://doi.org/10.1145/3569468>.
- [23] C. M. Bhushan, P. Koppuravuri, N. Prasanthi, F. Gazi, M. M. Hussain, M. Abdussami, A. A. Devi, and J. Faizi, "Deploying TinyML for energy-efficient object detection and communication in low-power edge AI systems," *Sci. Rep.*, vol. 15, no. 1, 2025, <https://doi.org/10.1038/s41598-025-27818-9>.
- [24] J. Moosmann, M. Giordano, C. Vogt, and M. Magno, "TinyissimoYOLO: A quantized, low-memory footprint, TinyML object detection network for low power microcontrollers," in *Proc. 2023 IEEE 5th Int. Conf. Artificial Intelligence Circuits and Systems (AICAS)*, Hangzhou, China, 2023, pp. 285–289, <https://doi.org/10.1109/AICAS57966.2023.10168657>.
- [25] YY. Abadade, N. Benamar, M. Bagaa, and H. Chaoui, "Empowering healthcare: TinyML for precise lung disease classification," *Future Internet*, vol. 16, no. 11, 2024, <https://doi.org/10.3390/fi16110391>.
- [26] A. Tschand, A. T. R. Rajan, S. Idgunji, A. Ghosh, J. Holleman, C. Király, P. Ambalkar, R. Borkar, R. Chukka, T. Cockrell, O. Curtis, G. Fursin, M. Hodak, H. Kassa, A. Lokhmotov, D. Miskovic, Y. Pan, M. P. Manmathan, L. Raymond, T. St. John, A. Suresh, R. Taubitz, S. Zhan, S. Wasson, D. Kanter, and V. J. Reddi, "MLPerf Power: Benchmarking the energy efficiency of machine learning systems from μ Watts to MWatts for sustainable AI," in *Proc. 2025 IEEE Int. Symp. High-Performance Computer Architecture (HPCA)*, Las Vegas, NV, USA, 2025, pp. 1201–1216, <https://doi.org/10.1109/HPCA61900.2025.00092>.
- [27] A. Sabovic, M. Aernouts, D. Subotic, J. Fontaine, E. De Poorter, and J. Famaey, "Towards energy-aware TinyML on battery-less IoT devices," *Internet Things*, vol. 22, Art. no. 100736, 2023, <https://doi.org/10.1016/j.iot.2023.100736>.

- [28] S. Blattmann, "Smoke detection dataset: real-time smoke detection with AI-based sensor fusion," public multi-sensor fire and smoke dataset, 2022. [Online]. Available: <https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset>.
- [29] LoRa Alliance Technical Committee, "RP002-1.0.4 LoRaWAN regional parameters," LoRa Alliance, 2022. [Online]. Available: https://lora-alliance.org/resource_hub/rp2-1-0-4-regional-parameters/.
- [30] Semtech Corporation, "SX1261/2 long range, low power, sub-GHz RF transceiver — datasheet," Rev. 2.1, 2021. [Online]. Available: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1262>.