Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Leakage-Aware Evaluation of Fault-State Detectors for a Low-Cost ESP32 Multi-Sensor Battery Safety Monitor

Abdul Hasib

sm.abdulhasib.bd@gmail.com

University of Frontier Technology, Bangladesh

Majedul Islam Naeem

Leading University

Nihal Das Ankur

Southeast University

Research Article

Keywords: Battery safety monitoring, ESP32, Data leakage, Blocked cross-validation, Embedded machine learning, Dataset audit

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11578110/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Leakage-Aware Evaluation of Fault-State Detectors for a Low-Cost ESP32 Multi-Sensor Battery Safety Monitor

Abdul Hasib[1*], Majedul Islam Naeem[2] and Nihal Das Ankur[3]

[1*]Department of Internet of Things and Robotics Engineering, University of Frontier Technology, Bangladesh.
[2]Department of Electrical and Electronic Engineering, Leading University, Sylhet, Bangladesh.
[3]Department of Electrical and Electronics Engineering, Southeast University, Bangladesh.

*Corresponding author(s). E-mail(s): sm.abdulhasib.bd@gmail.com;
Contributing authors: majedulnaeem@gmail.com;
nihaldas007@gmail.com;

Abstract

Low-cost microcontroller monitors for lithium-ion battery safety are frequently reported with near-perfect machine-learning accuracy, yet the datasets behind such figures are seldom audited and are usually split at random. This study documents an ESP32 prototype with a component cost below 2,000 BDT (16.19 USD) that samples cell voltage, motor-branch current, cell temperature and MQ-2 gas-sensor output every 2 s, applies a three-level threshold rule and drives a protection relay, and audits the 5,186-row dataset recorded on it. The dataset joins four phase recordings, each carrying a single label. The firmware rule reproduces 97.13% of the labels, temperature alone separates the three states, only 24.7% of temperature values fall on the output grid of the DS18B20 sensor, and 48.7% of current samples are positive in a branch that should conduct in one direction. Eight detectors (the firmware rule, five tabular learners, a one-dimensional convolutional network and a long short-term memory network) were compared under random, blocked and blocked five-fold splits with a 60-row embargo over five seeds. With temperature and gas available, every detector reached at least 0.994 macro-F1 under the blocked split, so no learner improved on the 126-byte rule. With voltage and current only, random splitting raised macro-F1 by 0.183 to 0.271 over blocked splitting (paired t-test, $p \leq 0.002$). Compiled for the ESP32,

the trained tabular detectors occupied 205 to 29,256 bytes. Recording-aware splits and a dataset audit should precede any accuracy claim for such monitors.

Keywords: Battery safety monitoring, ESP32, Data leakage, Blocked cross-validation, Embedded machine learning, Dataset audit

1 Introduction

Lithium-ion cells can enter thermal runaway, a chain of exothermic decomposition reactions that ends in venting, fire or explosion (Feng et al., 2018; Wang et al., 2012). Most fires in electric vehicles are traced to this failure of the battery (Sun, Bisschop, Niu, & Huang, 2020), and a battery management system is expected to detect abnormal voltage, current and temperature before the chain becomes self-sustaining (Lu, Han, Li, Hua, & Ouyang, 2013). Because electrolyte decomposition releases gases such as H_2, CO and CO_2 before or during the first venting (Koch, Fill, & Birke, 2018), gas sensing has been proposed as an earlier indicator than temperature or voltage alone (Cai et al., 2021; Essl, Seifert, Rabe, & Fuchs, 2021; Jin et al., 2020). A survey of monitoring and detection methods compares the sensitivity of these signals (Liao, Zhang, Li, Zhang, & Habetler, 2019).

These findings have encouraged low-cost monitors that pair a Wi-Fi microcontroller with off-the-shelf voltage, current, temperature and metal-oxide gas modules, and a growing number of studies place a machine-learning detector on top of such sensors. Reported accuracies are often close to perfect; for example, an Optuna-tuned XGBoost classifier for thermal-runaway categories reached 100% accuracy (El Abed, Nassreddine, Al-Khatib, Nassereddine, & Hellany, 2025). Scores of this kind depend on how the evaluation data were produced and split. When consecutive samples from a slowly varying recording are shuffled into both training and test sets, a model can match a test row to its near-identical neighbours instead of learning the state, a form of leakage documented for time series, structured data and windowed sensor streams (Bergmeir & Benítez, 2012; Hammerla & Plötz, 2015; Kaufman, Rosset, Perlich, & Stitelman, 2012; Roberts et al., 2017). A survey of reviews across 17 scientific fields found leakage of this and related kinds affecting 294 papers (Kapoor & Narayanan, 2023). Battery-monitor studies seldom report the provenance of their labels, whether the classes were recorded in separate sessions, or whether the learned detector improves on the threshold rule already running in the firmware.

The gap addressed here is the absence of an end-to-end, leakage-aware account of a low-cost battery safety monitor that links the hardware and firmware to the recorded dataset, the evaluation protocol and the cost of deploying each detector on the same microcontroller. The study makes three contributions:

1. A documented ESP32 multi-sensor prototype, including its schematic, pin map, firmware logic, telemetry path and component cost, together with a design review that confirms the relay disconnects the load at the critical state and identifies the conditions under which protection is not applied and the current channel does not describe the cell.

2. A reproducible audit of the 5,186-row dataset recorded on the prototype, which establishes that the three states are separable by temperature alone, that the labels follow the firmware rule in 97.13% of rows, and that several signal statistics are inconsistent with unprocessed sensor output.
3. A leakage-aware benchmark of eight detectors under three split protocols and seven feature groups, with paired significance tests and compiled ESP32 code size, showing that random splitting inflates macro-F1 by 0.183 to 0.271 when temperature and gas are withheld and that no learned detector outperforms the 126-byte firmware rule when they are present.

2 Related work

Gas and multi-signal detection of battery failure.

Laboratory studies have characterised the gases released before and during thermal runaway. An autoclave study of 51 automotive cells quantified vent-gas volume and composition and their dependence on energy density and capacity (Koch et al., 2018). Tests of several commercial gas sensors in four failure cases showed that detection with gas sensors is possible but depends strongly on the failure case (Essl et al., 2021). CO_2 sensors were used to detect cell failure and venting (Cai et al., 2021), H_2 capture flagged micro-scale lithium dendrites before smoke or fire in an overcharge experiment on an 8.8 kWh pack (Jin et al., 2020), and modelling of first-venting gas diffusion was used to size and place warning sensors in different pack layouts (Huang et al., 2023). A distributed array of CO sensors, including the MQ-7 metal-oxide module, was used to locate leakage sources inside a battery pack (Mateev, Marinova, & Kartunov, 2019). These works establish the physical value of gas sensing but evaluate sensors or detection criteria on controlled abuse tests rather than learned detectors on low-cost prototype data.

Data-driven fault and thermal-runaway warning.

Field data from electric vehicles have supported statistical and learned detectors. Big-data statistics and a neural network were combined to diagnose battery faults over a year of vehicle operation (Zhao, Liu, Wang, Zhang, & Hong, 2017). A long short-term memory network predicted voltage abnormalities from operational data of an electric taxi and was evaluated with 10-fold cross-validation (Hong, Wang, & Yao, 2019). A time–frequency attention model achieved 8 to 13 min of advance warning on real-world vehicle data (Ma, Huo, Wang, & Zhang, 2023), and a combination of model-driven and data-driven reasoning produced a 25 min advance warning with fewer false alarms (Q. Chen, He, Fang, & Yu, 2024). A cloud framework recognised mechanical-abuse failure modes and predicted short circuits from multi-source signals (Zhang, Pan, Cao, Liu, & Yu, 2024), an XGBoost classifier trained on more than 210 open-source abuse tests reached an F1-score of 0.98 on 42 independent tests (Alhasan & Alrashdan, 2026), and a deep residual network was proposed for detecting faulty battery-sensor data (Kosuru & Kavasseri Venkitaraman, 2023). The evaluation of Alhasan and Alrashdan (2026) keeps whole tests out of training, which is the safeguard that random row-level splits omit.

Embedded machine learning.

TinyML frameworks run inference on microcontrollers with kilobytes of memory (Ray, 2022). TensorFlow Lite Micro provides an interpreter for quantised networks (David et al., 2021), integer-only quantisation reduces memory roughly fourfold relative to 32-bit floating point (Jacob et al., 2018), and MLPerf Tiny standardises accuracy, latency and energy measurement for such systems (Banbury et al., 2021). For battery applications, temporal convolutional models for remaining useful life and cell temperature have been deployed on an Arduino Nano 33 BLE Sense (Weng, Guan, & Ababei, 2024). Tree ensembles and small networks can also be exported as plain C (Nordby, 2019), which makes code size a measurable property of each candidate detector.

Evaluation under dependence.

Random cross-validation underestimates error when observations share temporal, spatial or group structure, and blocked schemes are recommended instead (Bergmeir & Benítez, 2012; Roberts et al., 2017). In activity recognition, overlapping windows from the same session inflate cross-validated accuracy (Hammerla & Plötz, 2015). More generally, models exploit shortcuts that hold in a benchmark but not in deployment (Geirhos et al., 2020), and dataset documentation is advocated so that such properties are visible to users (Gebru et al., 2021). Table 1 positions the present study against the closest prior work.

3 Prototype hardware, firmware and telemetry

Figure 1 summarises the prototype and the study built on it. The monitor is assembled on perforated board around an ESP32 DevKit, with a single 18650 lithium-ion cell as the monitored element and a TT gear motor as a switchable load (Fig. 3). Figure 2 shows the schematic, which groups the circuit into power, sensing, control and actuator units; in the original drawing the net label on pin D21 duplicated SCL, and it is shown here as SDA to match the I^2C wiring. Table 2 lists every component with its connection and role, and Table 3 groups the same components by function and gives their cost. All of them were bought for less than 2,000 BDT, which is 16.19 USD at the Bangladesh Bank weighted average rate of 123.55 BDT per USD on 18 October 2026 (Bangladesh Bank, 2026).

3.1 Firmware

The firmware (Fig. 4) initialises the serial port at 115,200 baud, the INA219 and DS18B20 libraries, the relay pin and an ADC attenuation of 11 dB, then blocks until the Wi-Fi connection is established. In the main loop, a reading is taken whenever at least 2,000 ms have elapsed since the previous one. The current is read from the INA219 in milliamperes, the temperature from the DS18B20 in degrees Celsius, and the MQ-2 output as a raw 12-bit ADC count G. The cell voltage is the mean of 20

Table 1: Positioning of this study against the closest prior work

Study	Signals and data	Detector and evaluation	Focus relative to this study
Koch et al. (2018)	Vent gas of 51 automotive cells; autoclave tests	Gas volume and composition analysis	Gas characterisation; no detector
Essl et al. (2021)	Commercial gas sensors; automotive cells, four failure cases	Sensor response per failure case	Sensor physics; no learned detector
Cai et al. (2021)	CO_2 sensors; failure and venting tests	Venting detection from CO_2	Dedicated gas sensing; no learned detector
Jin et al. (2020)	H_2 gas capture; overcharge of an 8.8 kWh pack	Gas-signal early warning	Dendrite detection; no learned detector
Huang et al. (2023)	21700-cell venting tests; thermal-runaway and gas-diffusion models	Sensor count and layout by simulation	Sensor placement; no learned detector
Mateev et al. (2019)	Distributed CO sensors incl. MQ-7	Leak-source reconstruction	Localisation, not state classification
Zhao et al. (2017)	Cell terminal voltages; one year of vehicle operating data	3σ multi-level screening and neural network	Field diagnosis; no embedded footprint
Hong et al. (2019)	Voltage; electric-taxi field data	LSTM prognosis; 10-fold cross-validation	Field data; no embedded footprint
Ma et al. (2023)	Voltage and temperature; real-world vehicle data	Time–frequency attention network; 8–13 min warning	Field warning; no embedded footprint
Q. Chen et al. (2024)	Battery temporal state data	K-Means outliers with a Bernardi thermal model; 25 min warning	Hybrid model; no embedded footprint
Zhang et al. (2024)	Multi-source signals; mechanical abuse	SOM-BP failure-mode recognition; SVR prediction	Cloud framework; mechanical abuse only
El Abed et al. (2025)	Thermal-runaway cause data; generative oversampling	SVM, MLP, TabNet, XGBoost tuned with Optuna; three-fold cross-validation	Cause classification; no embedded footprint
Alhasan and Alrashdan (2026)	Voltage, temperature, force; 210+ open abuse tests	XGBoost; 42 held-out tests	Test-level hold-out on public abuse data
Kosuru and Kavasseri Venkitaraman (2023)	Battery sensor and transmission data	Optimised deep residual network; MATLAB simulation	Sensor-fault detection, not battery state
Weng et al. (2024)	Public cycling datasets	TinyML TCN on a microcontroller	RUL and temperature, not fault state
This work	V, I, T, MQ-2 gas; ESP32 prototype, below 16.19 USD in components; four joined recordings	Rule and 7 learners; random, blocked and blocked 5-fold splits; ESP32 code size	Dataset audit, leakage quantification, deployment cost

ADC samples taken 2 ms apart and scaled by the divider factor,

$$V = k\,\frac{3.3}{4095}\,\frac{1}{20}\sum_{j=1}^{20} a_j, \qquad k = 6.012, \tag{1}$$

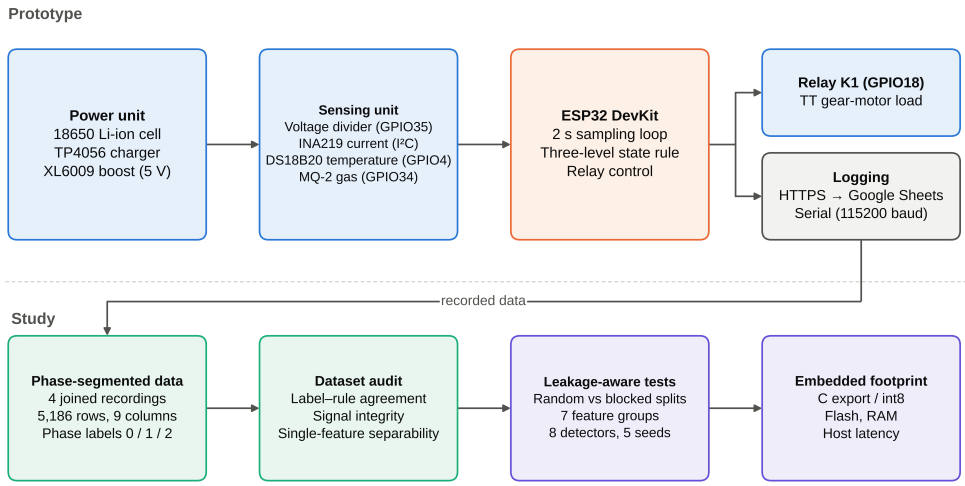


Fig. 1: Prototype architecture (top) and the study pipeline applied to its recorded data (bottom)

Table 2: Components, connections and roles in the prototype

Component	Connection	Role
ESP32 DevKit	Powered from the 5 V boost rail	Sampling, state rule, relay drive, Wi-Fi telemetry
18650 Li-ion cell	Cell holder; charged by TP4056	Monitored energy-storage element
TP4056 module	USB input; B+/B− to the cell	Single-cell charging
XL6009 boost module	Cell to 5 V via DIP switch	Supply for ESP32, relay module and MQ-2
Voltage-sensor module	Divider output to GPIO35 (ADC1)	Cell voltage V
INA219 module, 0.1 Ω shunt	I^2C, SDA GPIO21, SCL GPIO22; Vin− to relay K1 COM	Current I in the relay-switched motor branch
DS18B20	1-Wire on GPIO4, 4.7 kΩ pull-up	Cell temperature T
MQ-2 module	5 V supply; analogue output to GPIO34	Gas-sensor output G (ADC counts)
Two-channel relay module	IN1 on GPIO18; motor on K1 NO	Load disconnection at the critical state
TT gear motor	Between K1 NO and battery ground	Switchable load

where a_j is the j-th raw reading on GPIO35. The firmware then assigns a state s from temperature and gas alone,

$$s(T, G) = \begin{cases} 2, & T > 55\ ^\circ\text{C or } G > 350, \\ 1, & T > 40\ ^\circ\text{C or } G > 200, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where 0, 1 and 2 denote normal, warning and critical. GPIO18 is driven LOW when $s = 2$ and HIGH otherwise. Each reading is printed over the serial port and sent as

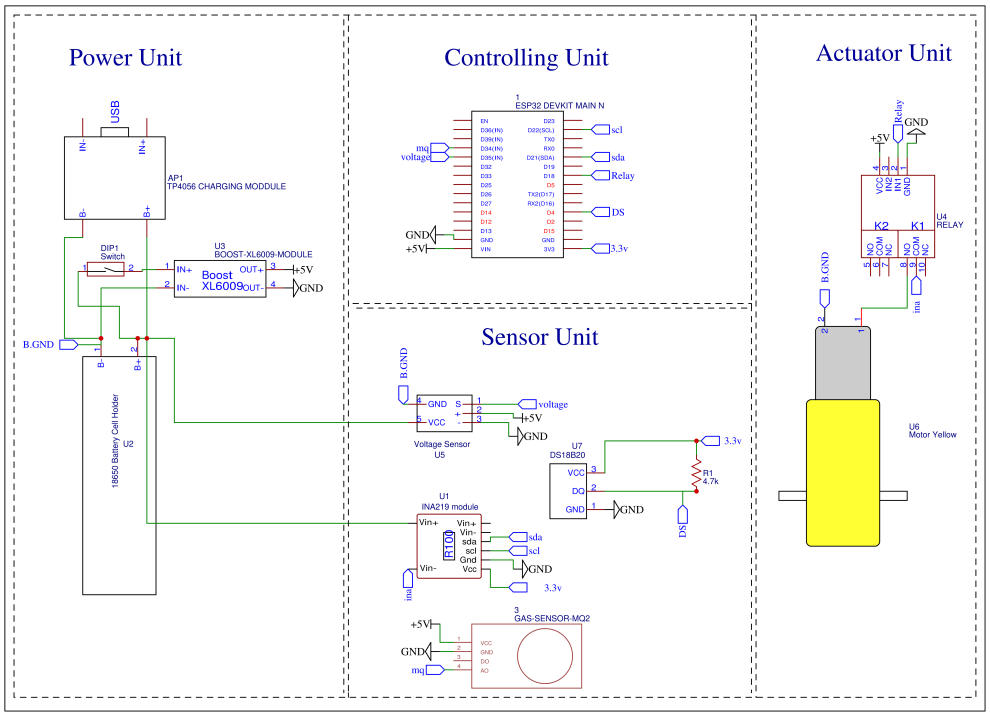


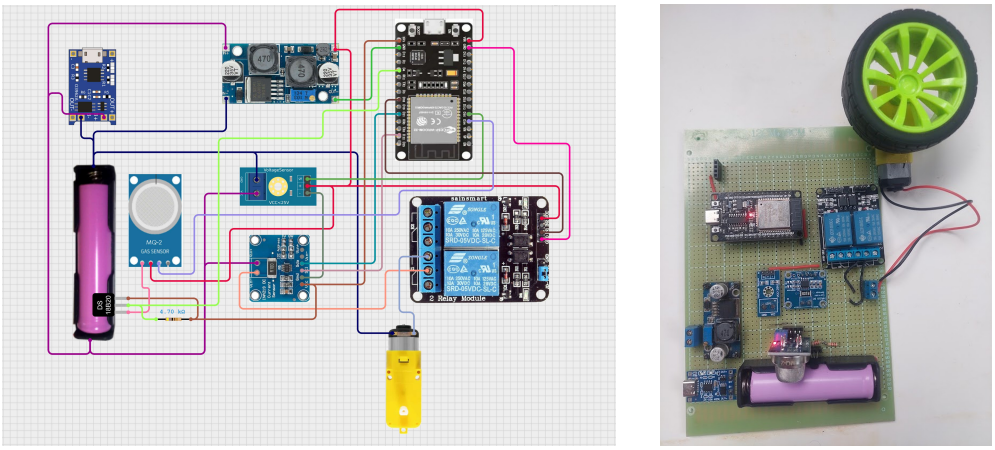
Fig. 2: Schematic of the monitor grouped into power, control, sensing and actuator units

Table 3: Component groups and total component cost of the prototype

Group	Components
Control	ESP32 DevKit; two-channel relay module
Sensing	Voltage-sensor module; INA219 module with 0.1 Ω shunt; DS18B20 with 4.7 kΩ pull-up; MQ-2 module
Power	18650 Li-ion cell and holder; TP4056 charger; XL6009 boost module; DIP switch
Load and assembly	TT gear motor; perforated board; wiring
Total component cost	Below 2,000 BDT (16.19 USD)

USD value at 123.55 BDT per USD, the Bangladesh Bank weighted average rate on 18 October 2026 (Bangladesh Bank, 2026).

an HTTPS GET request to a Google Apps Script web application that the firmware uses to log readings to a spreadsheet (Fig. 5). If Wi-Fi is lost, the firmware calls `WiFi.begin` again and continues sampling.



(a) (b)

Fig. 3: Wiring diagram (a) and assembled prototype (b)

3.2 Design review

Reading the schematic against the firmware identifies five properties that bound what the prototype can measure and protect. First, the protection rule runs only inside the main loop, and `setup()` does not return until Wi-Fi connects, so without a network the relay is never driven by the rule. Second, the relay input is driven LOW at the critical state and HIGH otherwise, and the motor is wired to the normally open K1 contact. On the prototype the motor ran in the normal state and stopped when the critical state was reached, so the module energises its coil while IN1 is HIGH and the relay disconnects the load as intended. Because the load runs only while the coil is energised, a loss of the relay-module supply also disconnects it. Third, the INA219 measures the voltage across its shunt (Texas Instruments, 2015), and the shunt sits in the motor branch, not in the cell's charge–discharge path, so I is the load current of a single-direction branch rather than the cell current. Fourth, the MQ-2 module is supplied from the 5 V rail and its analogue output is connected to GPIO34 without level shifting in the schematic; the recorded maximum of 1,000 counts corresponds to about 0.81 V at the pin under the 3.3/4095 scaling used in Eq. (1), so the recorded data did not approach the ADC limit. Fifth, `WiFiClientSecure` is used with certificate verification disabled, and the HTTPS request is blocking, so the effective sampling interval is 2 s or the time taken by one reading and request, whichever is longer; the DS18B20 alone needs up to 750 ms for a 12-bit conversion (Maxim Integrated, n.d.).

4 Dataset and audit

4.1 Provenance and structure

The released file contains 5,186 rows and nine columns (Table 4). According to the authors' records, it was recorded on this prototype as separate phase recordings that

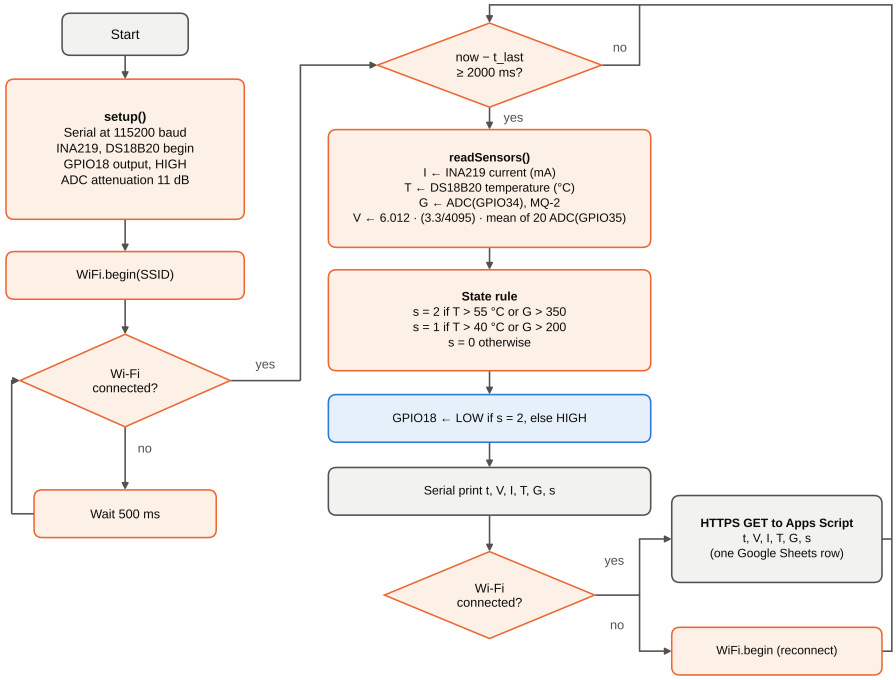


Fig. 4: Firmware control flow, including the blocking Wi-Fi wait in `setup()`

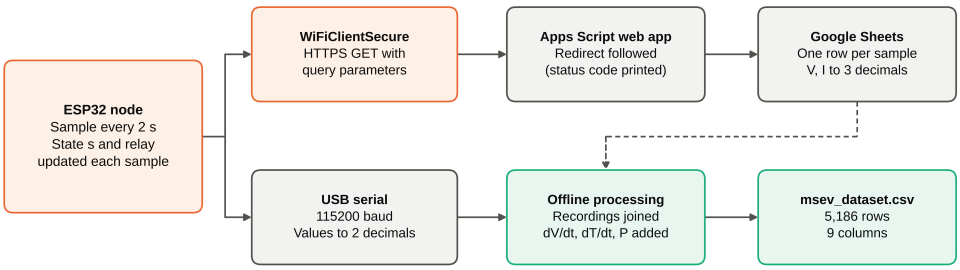


Fig. 5: Telemetry path from the ESP32 to the released dataset

were then joined, with one label assigned to each recording. The `Time` column is a row index with a step of exactly 1, not the firmware timestamp, so the 2 s interval is nominal. The derived columns were recomputed from the stored values:

$$\Delta V_t = V_t - V_{t-1}, \qquad \Delta T_t = T_t - T_{t-1}, \qquad P_t = V_t I_t, \tag{3}$$

Table 4: Columns of the released dataset

Column	Unit	Source
Time	sample index	Row index, step 1 (firmware interval 2 s)
Voltage	V	Measured, divider on GPIO35, Eq. (1)
Current	mA	Measured, INA219
Temperature	°C	Measured, DS18B20
Gas	ADC count (12-bit)	Measured, MQ-2 on GPIO34
Label	0/1/2	Phase label assigned per recording
dV_dt	V per sample	Derived offline, Eq. (3)
dT_dt	°C per sample	Derived offline, Eq. (3)
Power	mW	Derived offline, Eq. (3)

Table 5: Recording structure and the discontinuities at the three joins

Recording	Rows	Count	Label	T start → end (°C)	V start → end (V)
R1	0–1699	1,700	0	28.91 → 30.92	3.82 → 4.02
R2	1700–3399	1,700	1	39.32 → 45.17	3.69 → 3.41
R3	3400–4999	1,600	2	55.83 → 90.00	3.41 → 2.80
R4	5000–5185	186	2	57.62 → 63.17	3.49 → 3.36

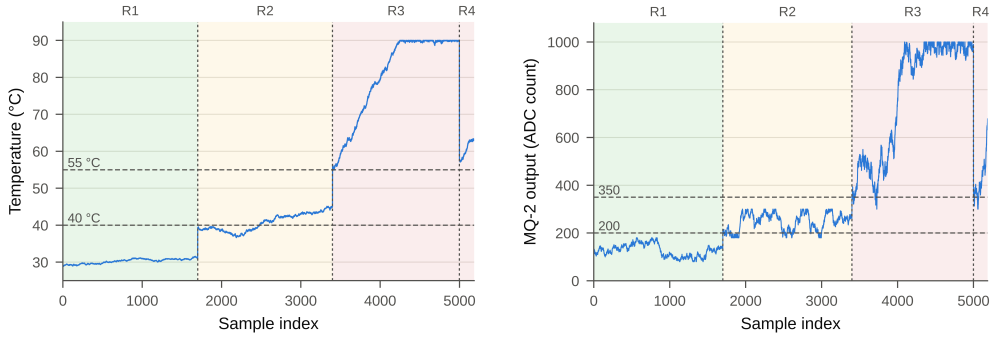
Join row	ΔT (°C)	ΔV (V)	Stored `dT_dt`	Stored `dV_dt`
1700	+8.40	-0.33	8.40	-0.33
3400	+10.66	+0.00	10.66	0.00
5000	-32.38	+0.69	0.00	0.00

Table 6: Per-class statistics, mean ± standard deviation and [minimum, maximum]

	Normal (0), n = 1,700		Warning (1), n = 1,700		Critical (2), n = 1,786	
Signal	Mean ± SD	Range	Mean ± SD	Range	Mean ± SD	Range
V (V)	3.94 ± 0.11	[3.70, 4.19]	3.57 ± 0.10	[3.40, 3.80]	3.14 ± 0.19	[2.80, 3.50]
I (mA)	-72.6 ± 324.9	[-468.2, 367.2]	-3.7 ± 258.6	[-896.5, 731.5]	4.5 ± 789.3	[-2197.2, 2093.1]
T (°C)	30.27 ± 0.64	[28.87, 31.57]	40.76 ± 2.34	[36.64, 45.17]	78.58 ± 12.48	[54.89, 90.00]
G (counts)	126 ± 26	[80, 180]	247 ± 36	[180, 300]	742 ± 253	[300, 1000]

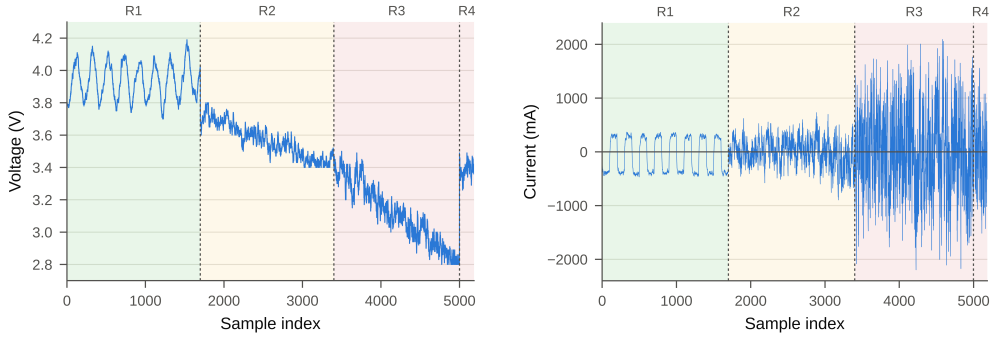
where ΔV_t and ΔT_t are stored as `dV_dt` and `dT_dt` in units per sample and P_t in milliwatts. P_t matched $V_t I_t$ to within 9.1×10^{-13}, and both differences matched Eq. (3) at every row except row 5,000, where they were reset to zero.

Discontinuities in temperature and voltage locate the joins at rows 1,700, 3,400 and 5,000, which yields the four recordings R1–R4 in Table 5. Each recording carries a single label: R1 is normal, R2 warning, and R3 and R4 critical. At the first two joins the stored temperature difference equals the jump between recordings (8.40 and 10.66 °C per sample); these two values are the largest `dT_dt` entries of their classes and are artefacts of joining rather than thermal events. Figures 6 and 7 show the four signals with the recordings shaded, and Table 6 gives their statistics per class.



(a) (b)

Fig. 6: Temperature (a) and MQ-2 output (b) over the joined recordings, with the firmware thresholds dashed



(a) (b)

Fig. 7: Cell voltage (a) and motor-branch current (b) over the joined recordings

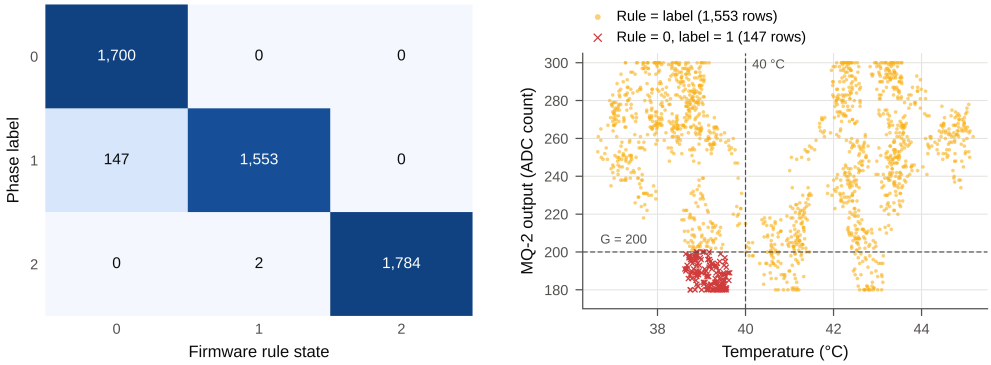
4.2 Audit procedure

The audit tested four properties that decide what any detector trained on the file can learn. Label provenance was measured by applying Eq. (2) to every row and comparing s with the stored label. Class separability was measured by the five-seed accuracy of a depth-2 decision tree trained on each feature alone. Signal integrity was assessed with three statistics. The first is the fraction of temperature values whose two-decimal rendering lies on the output grid of the DS18B20, whose readings are multiples of 0.0625 °C at 12-bit resolution and of coarser steps at lower resolutions (Maxim Integrated, n.d.),

$$\gamma = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[T_i - \lfloor T_i \rfloor \in \mathcal{G} \right], \tag{4}$$

Table 7: Dataset audit results

Check	Result
Rows / columns	5,186 / 9
Missing values; duplicate rows	0; 0
Time step	1 (row index)
Joins between recordings (rows)	1,700; 3,400; 5,000
$P = VI$, maximum absolute error	9.1×10^{-13}
Derived differences matching Eq. (3)	All rows except row 5,000 (reset at join)
Label equal to firmware rule, Eq. (2)	5,037 / 5,186 (97.13%)
Temperature on DS18B20 grid, γ in Eq. (4)	24.7% (1,281 rows; 314 of them at 90.00 °C)
Distinct hundredths in temperature	100 of 100
Rows at $T = 90.00$ °C; at $G = 1{,}000$	314; 157
Positive current samples	48.7%



(a) (b)

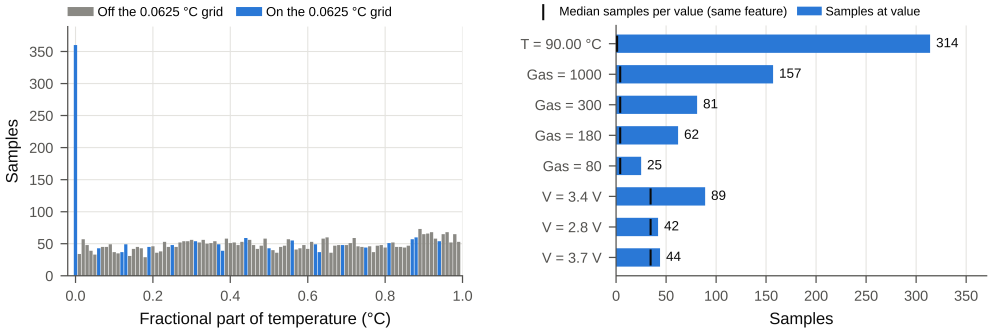
Fig. 8: Firmware rule against the stored labels (a) and the warning rows it calls normal, all below both thresholds (b)

where $\mathcal{G}$ contains the 16 two-decimal renderings of $0.0625m$, $m = 0, \ldots, 15$, with both roundings admitted at the four half-cent cases, giving 20 admissible values. Under a uniform distribution of hundredths, γ would be 0.20. The second statistic counts repeated values at the extremes of each class range, and the third is the fraction of positive current samples in a branch that can only conduct in one direction.

4.3 Audit results

Table 7 lists the outcomes. The file has no missing or duplicated rows. The firmware rule reproduces 5,037 of the 5,186 labels (97.13%; Fig. 8a). All 149 disagreements are under-calls: 147 warning rows at the start of R2, where the temperature is still below 40 °C and the gas output below 200 counts (Fig. 8b), and 2 critical rows early in R3. The labels therefore encode the recording a row came from, and the recordings were chosen to follow the rule.

The class ranges are disjoint in temperature: normal rows span 28.87–31.57 °C, warning rows 36.64–45.17 °C and critical rows 54.89–90.00 °C (Table 6). A depth-2



(a) (b)

Fig. 9: Hundredths of the temperature values against the DS18B20 grid (a) and pile-ups at class-range extremes (b)

tree on temperature alone classifies every row correctly, gas alone reaches 0.996 and voltage 0.948, whereas current, power and both differences stay between 0.464 and 0.528 against a chance level of 0.333 (Fig. 10b). Temperature and gas are strongly correlated with each other ($r = 0.97$) and with voltage ($r = -0.93$ and -0.90), while current is correlated only with power (Fig. 10a).

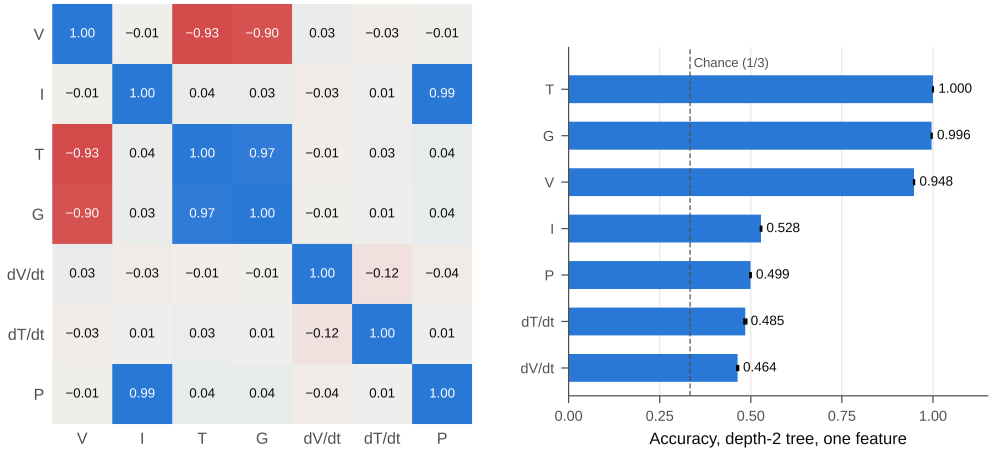
Three statistics suggest that the stored values are not raw sensor output. Only 24.7% of temperatures fall on the DS18B20 grid, close to the 20% expected by chance, and all 100 possible hundredths occur (Fig. 9a); readings from this sensor at any resolution would lie on the grid. Values accumulate at round numbers at the edges of the class ranges: 314 rows equal 90.00 °C and 157 equal 1,000 gas counts, both below the sensors' limits, while the median number of rows per distinct value is 1 for temperature and 4 for gas (Fig. 9b). The gas class ranges meet exactly at 180 and 300 counts, where 62 and 81 rows sit. Finally, 48.7% of current samples are positive and 51.3% negative in a branch that conducts in one direction, and the sign changes between consecutive samples in 0.9%, 22.5% and 30.1% of normal, warning and critical rows. These patterns are consistent with clipping, rescaling or other processing between acquisition and release; the project files do not document such a step, and the audit cannot identify it.

5 Leakage-aware benchmark

5.1 Split protocols

Three protocols were compared. Protocol S1 is the common practice of a stratified random 70/30 split of rows. Protocol S2 is a blocked 70/30 split applied inside each recording r with n_r rows: the first $\lfloor 0.7 n_r \rfloor - E$ rows form the training set, the last $n_r - \lfloor 0.7 n_r \rfloor$ rows form the test set, and the $E = 60$ rows between them are discarded as an embargo,

$$\mathcal{D}_r^{\mathrm{tr}} = \{1, \ldots, \lfloor 0.7 n_r \rfloor - E\}, \qquad \mathcal{D}_r^{\mathrm{te}} = \{\lfloor 0.7 n_r \rfloor + 1, \ldots, n_r\}. \tag{5}$$



(a) (b)

Fig. 10: Pearson correlation between features (a) and accuracy of a depth-2 tree on each feature alone (b)

Protocol S3 divides each recording into five contiguous blocks; fold k tests block k and trains on the rows more than E rows away from it on either side. Splitting inside each recording keeps every class in both sets, and the embargo of 60 rows (120 s nominal) separates training and test rows that would otherwise be adjacent. S1 and S2 were repeated over five seeds, and S3 over its five folds.

All detectors are scored on the same rows. A test row t is scored only if the window of $L = 20$ rows ending at t lies inside one recording; under S2 and S3 the window must also lie inside one contiguous test segment,

$$\mathcal{E} = \{t \in \mathcal{D}^{\text{te}} : t - L + 1, \ldots, t \in \text{the same test segment}\}. \tag{6}$$

Under S1, windows may contain training rows, which reproduces the naive practice being measured. Scalers are fitted on training rows only, and hyperparameters are chosen on an inner validation set that follows the outer protocol: a random 20% of training rows for S1 and the last 20% of each recording's training rows for S2 and S3. Algorithm 1 summarises the procedure.

5.2 Detectors and feature groups

Eight detectors were compared (Table 8): the firmware rule of Eq. (2), logistic regression (LR), a decision tree (DT), a random forest (RF) (Breiman, 2001), gradient-boosted trees (XGB) (T. Chen & Guestrin, 2016), a multilayer perceptron (MLP), a one-dimensional convolutional network (1D-CNN) and a long short-term memory network (LSTM) (Hochreiter & Schmidhuber, 1997). The tabular models were implemented with scikit-learn (Pedregosa et al., 2011) and see one row at a time; the two sequence models see the window of $L = 20$ rows ending at the scored row. The grids

Algorithm 1 Leakage-aware evaluation of one detector

Require: rows with recording id r, label y; protocol $\in \{\text{S1}, \text{S2}, \text{S3}\}$; embargo E; window L; candidate settings Θ

for each seed (S1, S2) or fold (S3) **do**
  if protocol is S1 **then**
    draw a stratified random 70/30 split of rows
  else
    split each recording into contiguous training and test blocks, Eq. (5); drop E rows at each boundary
  end if
  form the scoring set $\mathcal{E}$ of Eq. (6)
  split training rows into fit and validation parts using the same protocol
  for each $\theta \in \Theta$ **do**
    fit scaler and model on the fit part; record validation macro-F1
  end for
  refit the best θ on all training rows (tabular models) and predict on $\mathcal{E}$
  compute macro-F1, per-class recall and false-alarm rate on $\mathcal{E}$
end for
report mean and standard deviation; test S1 against S2 with a paired t-test over seeds

Table 8: Detector configurations and hyperparameter grids, selected on the inner validation set

Detector	Configuration
Rule	Eq. (2); no training
LR	Inverse regularisation $C \in \{0.1, 1, 10\}$; up to 3,000 iterations
DT	Maximum depth $\in \{3, 5, 8\}$
RF	200 trees; maximum depth $\in \{8, \text{unlimited}\}$
XGB	200 trees; learning rate 0.1; row and column subsampling 0.9; maximum depth $\in \{3, 6\}$
MLP	Hidden layers (64, 32); L2 penalty $\in \{10^{-4}, 10^{-2}\}$; early stopping; up to 400 iterations
1D-CNN	Two 16-filter convolutions of width 3, global average pooling, 16-unit dense layer; Adam, learning rate 10^{-3}; batch 64; up to 40 epochs, patience 5
LSTM	One 16-unit LSTM layer, 16-unit dense layer; training as for 1D-CNN

were kept small because the question is how evaluation changes the ranking, not how far tuning can push a score. Seven feature groups were used: all seven numerical columns; the four raw signals V, I, T, G; T alone; G alone; V and I; V, I, ΔV and ΔT; and I, P, ΔV and ΔT. The groups without T and G ask whether the electrical signals carry the state.

5.3 Metrics and statistics

The primary metric is macro-F1 over the three states (Sokolova & Lapalme, 2009), defined in Eq. (7),

$$F_1^{\text{macro}} = \frac{1}{3} \sum_{c=0}^{2} \frac{2\, P_c\, R_c}{P_c + R_c}, \tag{7}$$

where P_c and R_c are the precision and recall of state c. Per-class recall and the false-alarm rate of Eq. (8), the share of normal rows raised to warning or critical,

$$\mathrm{FAR} = \frac{|\{t \in \mathcal{E} : y_t = 0,\ \hat{y}_t \geq 1\}|}{|\{t \in \mathcal{E} : y_t = 0\}|}, \tag{8}$$

are reported alongside it. For each detector and feature group, S1 and S2 were compared with the paired t-test of Eq. (9) over the five seeds,

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}, \qquad d_i = F_{1,i}^{\mathrm{S1}} - F_{1,i}^{\mathrm{S2}}, \quad n = 5, \tag{9}$$

where $\bar{d}$ and s_d are the mean and standard deviation of the paired differences. Under S2 the split itself does not change with the seed, so the seed only varies model initialisation and subsampling. For RF and XGB on all seven features, permutation importance was measured as the mean drop in test macro-F1 over 10 random permutations of each feature.

5.4 Embedded deployment

The footprint was measured for the four raw signals under S2 with seed 0 and the hyperparameters selected there. DT, RF and MLP were exported to C with emlearn (Nordby, 2019), LR with m2cgen, and XGB with a custom exporter that writes each tree as nested comparisons. The rule was written by hand. Each C file was compiled for the ESP32 with `xtensa-esp32-elf-gcc -Os`, and flash use was taken as the sum of the text, literal, read-only data and initialised-data sections of the object file, with static RAM as initialised plus zero-initialised data. The same sources were compiled for the host with `gcc -O2` to verify that the C predictions match the Python models and to time single-sample inference. The 1D-CNN and LSTM were converted to full-integer TensorFlow Lite models (Jacob et al., 2018); their size is that of the int8 flatbuffer and excludes the TensorFlow Lite Micro runtime (David et al., 2021), and their latency was measured with the TensorFlow Lite interpreter on the host. Host latencies rank the detectors but are not ESP32 timings.

6 Results

6.1 Split protocol and feature group

Table 9 reports macro-F1 for the four raw signals and for voltage and current alone, and Fig. 11 plots the same results. With the four raw signals, every detector reached at least 0.994 under S2, and the rule scored 1.000 (Fig. 11a). The rule's lower S1 score of 0.970 arises because random test rows include the early-R2 rows that it calls normal, whereas the S2 test blocks come from late in each recording. S3 exposes the same rows: in the fold that holds out the first block of every recording, the rule's macro-F1 fell to 0.839 with a warning recall of 0.548, and the learned detectors also lost accuracy there.

With voltage and current alone, the protocols diverge (Fig. 11b). Under S1 the detectors scored 0.952 to 1.000, and the two sequence models, whose windows may

Table 9: Macro-F1 (mean ± SD over five seeds or folds) and the S1−S2 difference for V and I with its paired t-test p-value

	V, I, T, G			V, I only			
Detector	S1	S2	S3	S1	S2	S3	S1−S2 (p)
Rule	0.970±0.003	1.000±0.000	0.968±0.072	–	–	–	–
LR	0.999±0.001	1.000±0.000	0.972±0.062	0.952±0.008	0.719±0.000	0.876±0.121	+0.233 (3.1e−7)
DT	1.000±0.000	1.000±0.000	0.996±0.009	0.966±0.002	0.783±0.000	0.924±0.065	+0.183 (5.8e−9)
RF	1.000±0.000	1.000±0.000	0.996±0.009	0.966±0.004	0.749±0.004	0.913±0.084	+0.217 (2.4e−7)
XGB	1.000±0.000	0.998±0.002	0.918±0.136	0.966±0.004	0.736±0.002	0.916±0.078	+0.230 (8.1e−8)
MLP	0.999±0.001	0.999±0.003	0.963±0.083	0.961±0.005	0.774±0.016	0.908±0.089	+0.187 (9.2e−6)
1D-CNN	1.000±0.000	1.000±0.000	0.969±0.069	1.000±0.000	0.774±0.019	0.884±0.126	+0.226 (1.1e−5)
LSTM	1.000±0.000	0.994±0.005	0.977±0.033	0.999±0.001	0.729±0.084	0.870±0.142	+0.271 (2.0e−3)

The rule uses T and G only. S1: random 70/30; S2: blocked 70/30 with a 60-row embargo; S3: blocked 5-fold with the same embargo.

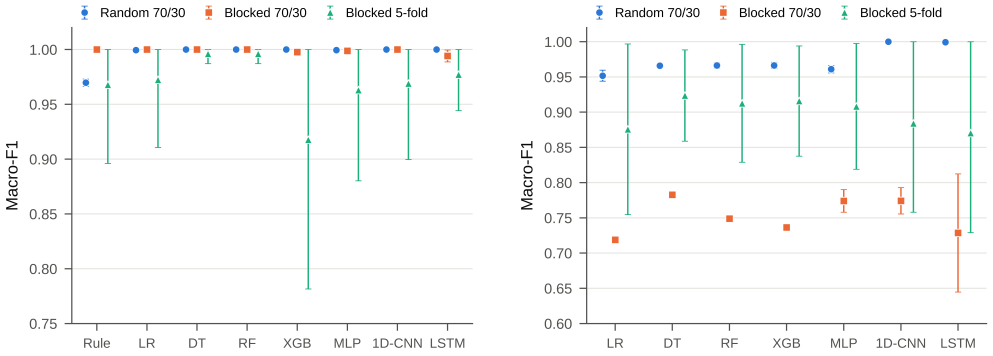
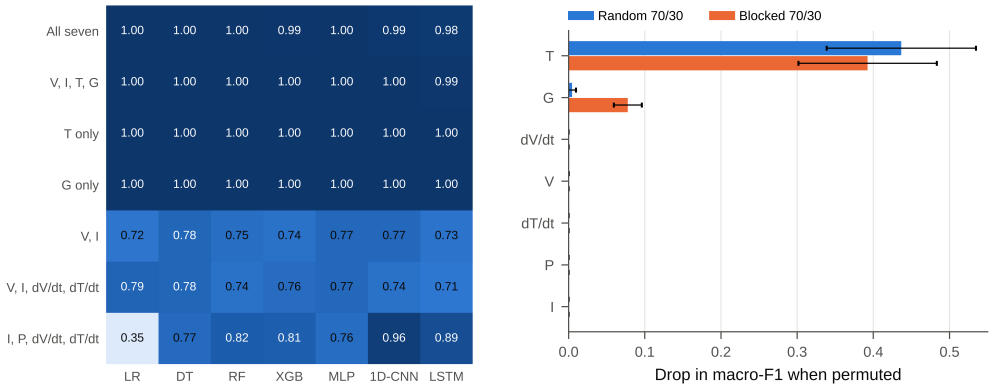


Fig. 11: Macro-F1 under the three protocols with V, I, T, G (a) and with V, I only (b); bars show one standard deviation

overlap training rows, reached 1.000 and 0.999. Under S2 the same detectors scored 0.719 to 0.783. The S1−S2 differences ranged from 0.183 for DT to 0.271 for LSTM, all with $p \leq 0.002$. Adding ΔV and ΔT did not close the gap: under S2 the scores were 0.711 to 0.793, with S1−S2 differences of 0.161 to 0.288 ($p \leq 1.4 \times 10^{-4}$). The failure under S2 is concentrated in the warning state. With voltage and current, warning recall was 0.295 to 0.446 while critical recall stayed at 0.992 or above, and false-alarm rates were 0.035 to 0.055. S3 scores for voltage and current lay between the two (0.870 to 0.924) with standard deviations of 0.065 to 0.142, showing that the result depends on which block is held out.

Figure 12a summarises the S2 results for all seven feature groups. Every group that contains T or G scored at least 0.979 for every learner, and T alone or G alone gave 1.000 for every learner. Without them, scores fell to 0.711–0.793 for the voltage–current groups and to 0.348–0.961 for current, power and both differences, where the 1D-CNN was the strongest at 0.961. For T alone, S3 revealed instability in two



(a) (b)

Fig. 12: Macro-F1 under S2 for each feature group and detector (a) and permutation importance for RF and XGB on all seven features (b)

learners: XGB scored 0.847±0.214 and MLP 0.941±0.130, because in single folds their warning recall fell to between 0.047 and 0.249, while DT, RF and LSTM scored 1.000 in every fold. Gas alone scored slightly higher under S2 (1.000 for every learner) than under S1 (0.991 to 1.000). Permutation importance confirms that the trees rely on temperature and gas (Fig. 12b). Permuting temperature reduced macro-F1 by 0.516 on average under S1 and 0.427 under S2, permuting gas by 0.003 and 0.117, and every other feature by less than 0.006.

6.2 Embedded footprint

Table 10 and Fig. 13 report the deployment cost. The rule compiled to 126 bytes, DT to 205 bytes and LR to 745 bytes, and these three scored 1.000 in C. RF needed 29,256 bytes, XGB 19,775 bytes and MLP 11,283 bytes of flash plus 532 bytes of static RAM. The C and Python predictions agreed on all 1,480 scored rows for every tabular detector. The int8 1D-CNN flatbuffer occupied 7,976 bytes with a macro-F1 of 1.000; the unrolled LSTM occupied 93,304 bytes, its int8 predictions agreed with the floating-point model on 99.66% of rows, and its macro-F1 was 0.984. On the host, single-sample inference took 0.0035 μs for the rule and 0.384 μs for XGB, and 13.9 and 35.2 μs per window for the interpreted 1D-CNN and LSTM. The smallest detectors were therefore also as accurate as any other on this dataset, so code size and accuracy did not trade off.

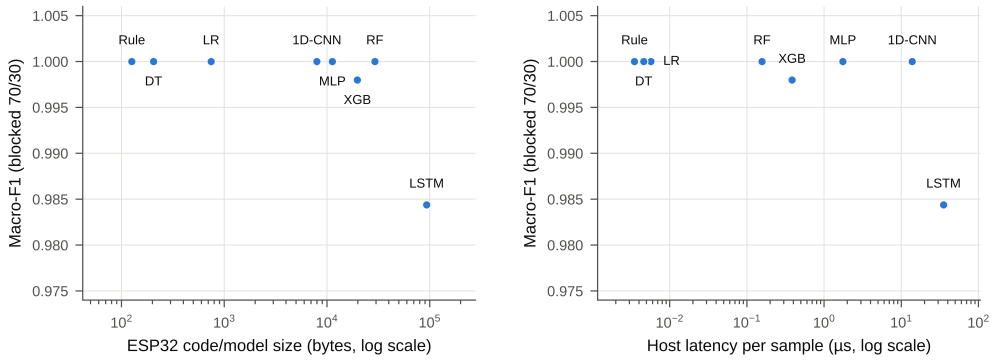
7 Discussion

The prototype dataset cannot show that a learned detector improves battery safety, because its labels were fixed by the recording in which each row was taken and, apart from the opening rows of R2 and R3, the recordings were set at temperatures and gas levels that already satisfy the firmware thresholds. Temperature alone separates

Table 10: Deployment cost for V, I, T, G under S2 (seed 0)

Detector	Setting	Parameters	Flash (B)	RAM (B)	Host latency (μs)	Agreement (%)	Macro-F1
Rule		4	126	0	0.0035	100.00	1.0000
LR	C=0.1	15	745	0	0.0057	100.00	1.0000
DT	max_depth=3	5	205	0	0.0046	100.00	1.0000
RF	max_depth=8	1,810	29,256	0	0.1570	100.00	1.0000
XGB	max_depth=3	1,894	19,775	0	0.3844	100.00	0.9980
MLP	alpha=0.0001	2,499	11,283	532	1.7489	100.00	1.0000
1D-CNN	L=20	1,315	7,976	–	13.8593	100.00	1.0000
LSTM	L=20	1,667	93,304	–	35.2387	99.66	0.9844

Parameters: thresholds (rule), coefficients (LR), tree nodes (DT, RF, XGB) or weights (MLP, 1D-CNN, LSTM). Flash for the 1D-CNN and LSTM is the int8 flatbuffer without the TensorFlow Lite Micro runtime. Agreement is between C or int8 predictions and the Python model.



(a) (b)

Fig. 13: Macro-F1 against ESP32 code or model size (a) and host latency per sample (b)

the classes with a gap of more than 5 °C between them, and the rule agrees with 97.13% of labels. Any learner given temperature or gas therefore reproduces Eq. (2), as the permutation importances show, and its near-perfect score measures the separation designed into the recordings rather than an ability to anticipate a fault. This is the situation described as shortcut learning (Geirhos et al., 2020): the benchmark is solvable, but the solution does not answer the deployment question. The fair baseline for any learned detector on such data is the rule already in the firmware, and on this dataset none of the seven learners exceeded it.

The voltage–current results show how much a random split can add when the state is not directly observable. Each recording has its own voltage level and drift, so under S1 a test row has training neighbours from the same recording, a few samples away and almost identical; the model can name the recording without learning the state, and windows that overlap training rows let the sequence models do this almost perfectly. Blocking the split by recording time and adding the embargo removes these neighbours, and the score falls by 0.183 to 0.271, mostly through the warning state,

whose voltage range overlaps both neighbouring classes. This matches the mechanism reported for windowed sensor data and structured cross-validation (Hammerla & Plötz, 2015; Roberts et al., 2017), and it implies that a macro-F1 near 1.0 from a random split of joined phase recordings should not be read as evidence that electrical signals predict battery state. The S3 variability adds a second caution: with only one recording per state, results hinge on which block is held out, and the opening block of each recording, nearest the phase change, is the hardest.

For a monitor whose components cost less than 16.19 USD, the footprint results favour the simplest detectors. The 126-byte rule and the 205-byte tree are at least 140 times smaller than RF and at least 450 times smaller than the LSTM, the 745-byte logistic model is about 39 times smaller than RF, and the unrolled LSTM needs 93,304 bytes before the interpreter is added while being the least accurate detector on these features. The larger risks to protection lie in the hardware and firmware rather than in the classifier. The relay disconnected the load at the critical state as intended, but protection must not wait for Wi-Fi, the current sensor must sit in the cell's own path if current is to describe the cell, and the gas input should be level-shifted to the ADC range. A dataset that could test early warning would need timestamps, raw sensor values, several independent sessions per state, and transitions recorded within a session, so that lead time can be measured against an event rather than against a label boundary.

8 Limitations and future work

The study rests on one prototype, one cell and one dataset of four joined recordings, and no thermal runaway, venting or abuse event was recorded, so none of the results measures detection lead time or real fault sensitivity. The processing that produced the off-grid temperatures, the value pile-ups and the sign-alternating current could not be identified from the project files, and raw serial logs or spreadsheet exports were not available to compare with the released file. The component cost is a total for the whole build at local prices in Bangladesh and is not itemised per component. Latencies were measured on a host computer rather than on the ESP32, and the network footprints exclude the inference runtime, so on-device timing, energy and total memory remain to be measured.

The next step follows directly from these gaps. Future recordings should keep the firmware timestamps and raw sensor values in the released file, repeat each state over several independent sessions, include transitions within sessions, and add a controlled heating or gas-release protocol under appropriate safety measures, so that recording-level hold-out and lead-time analysis become possible. The hardware revision should move protection ahead of the Wi-Fi connection, place the current sensor in the cell path and level-shift the gas input, after which the detectors in Table 10 can be timed on the ESP32 itself.

9 Conclusion

Low-cost battery safety monitors are often credited with near-perfect machine-learning accuracy on data whose provenance and splits are not examined. For an ESP32 monitor

of voltage, motor-branch current, temperature and gas with a component cost below 2,000 BDT (16.19 USD), an audit of its 5,186-row dataset showed that four single-label recordings separated by temperature determine the labels, which follow the firmware rule in 97.13% of rows. A leakage-aware benchmark of eight detectors then found that no learner surpassed the 126-byte rule when temperature and gas were available, and that random splitting raised macro-F1 by 0.183 to 0.271 when only voltage and current were used. Accuracy claims for such monitors therefore require recording-aware splits, a dataset audit and a comparison with the firmware rule before a learned detector is deployed.

Statements and Declarations

- **Funding:** The authors did not receive support from any organization for the submitted work.
- **Competing interests:** The authors have no relevant financial or non-financial interests to disclose.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** The dataset analysed during the current study is available from the corresponding author on reasonable request.
- **Code availability:** The analysis code is available from the corresponding author on reasonable request.
- **Author contributions:** Abdul Hasib: conceptualization, hardware and firmware development, data curation, formal analysis, writing (original draft). Majedul Islam Naeem: investigation, validation, visualization, writing (review and editing). Nihal Das Ankur: investigation, validation, writing (review and editing). All authors read and approved the final manuscript.

References

Alhasan, I., & Alrashdan, M.H.S. (2026). XGBoost-powered predictive analytics for early identification of thermal runaway in lithium-ion batteries. *World Electric Vehicle Journal*, *17*(2), 68, https://doi.org/10.3390/wevj17020068

Banbury, C., Janapa Reddi, V., Torelli, P., Jeffries, N., Kiraly, C., Holleman, J., ... Xuesong, X. (2021). MLPerf Tiny benchmark. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* (Vol. 1).

Bangladesh Bank (2026). *Foreign exchange rate: Weighted average exchange rate as on 18 October 2026.* Bangladesh Bank, Dhaka. https://www.bb.org.bd/econdata/exchange_rate/exchange_rate_18_10_2026.pdf. Accessed 30 October 2026.

Bergmeir, C., & Benítez, J.M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213, https://doi.org/10.1016/j.ins.2011.12.028

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32, https://doi.org/10.1023/A:1010933404324

Cai, T., Valecha, P., Tran, V., Engle, B., Stefanopoulou, A., Siegel, J. (2021). Detection of Li-ion battery failure and venting with carbon dioxide sensors. *eTransportation*, *7*, 100100, https://doi.org/10.1016/j.etran.2020.100100

Chen, Q., He, Y., Fang, N., Yu, G. (2024). A combined data-driven and model-based algorithm for accurate battery thermal runaway warning. *Sensors*, *24*(15), 4964, https://doi.org/10.3390/s24154964

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., ... Rhodes, R. (2021). TensorFlow Lite Micro: Embedded machine learning for TinyML systems. *Proceedings of Machine Learning and Systems* (Vol. 3).

El Abed, A., Nassreddine, G., Al-Khatib, O., Nassereddine, M., Hellany, A. (2025). Explainable and Optuna-optimized machine learning for battery thermal runaway prediction under class imbalance conditions. *Thermo*, *5*(3), 23, https://doi.org/10.3390/thermo5030023

Essl, C., Seifert, L., Rabe, M., Fuchs, A. (2021). Early detection of failing automotive batteries using gas sensors. *Batteries*, *7*(2), 25, https://doi.org/10.3390/batteries7020025

Feng, X., Ouyang, M., Liu, X., Lu, L., Xia, Y., He, X. (2018). Thermal runaway mechanism of lithium ion battery for electric vehicles: A review. *Energy Storage Materials*, *10*, 246–267, https://doi.org/10.1016/j.ensm.2017.05.013

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Daumé III, H., Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, *64*(12), 86–92, https://doi.org/10.1145/3458723

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, *2*(11), 665–673, https://doi.org/10.1038/s42256-020-00257-z

Hammerla, N.Y., & Plötz, T. (2015). Let's (not) stick together: Pairwise similarity biases cross-validation in activity recognition. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)* (pp. 1041–1051). ACM.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735

Hong, J., Wang, Z., Yao, Y. (2019). Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks. *Applied Energy*, *251*, 113381, https://doi.org/10.1016/j.apenergy.2019.113381

Huang, W., Feng, X., Pan, Y., Jin, C., Sun, J., Yao, J., ... Ouyang, M. (2023). Early warning of battery failure based on venting signal. *Journal of Energy Storage*, *59*, 106536, https://doi.org/10.1016/j.est.2022.106536

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2704–2713).

Jin, Y., Zheng, Z., Wei, D., Jiang, X., Lu, H., Sun, L., ... Cui, Y. (2020). Detection of micro-scale Li dendrite via H_2 gas capture for early safety warning. *Joule*, *4*(8), 1714–1729, https://doi.org/10.1016/j.joule.2020.05.016

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9), 100804, https://doi.org/10.1016/j.patter.2023.100804

Kaufman, S., Rosset, S., Perlich, C., Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, *6*(4), 15, https://doi.org/10.1145/2382577.2382579

Koch, S., Fill, A., Birke, K.P. (2018). Comprehensive gas analysis on large scale automotive lithium-ion cells in thermal runaway. *Journal of Power Sources*, *398*, 106–112, https://doi.org/10.1016/j.jpowsour.2018.07.051

Kosuru, V.S.R., & Kavasseri Venkitaraman, A. (2023). A smart battery management system for electric vehicles using deep learning-based sensor fault detection. *World Electric Vehicle Journal*, *14*(4), 101, https://doi.org/10.3390/wevj14040101

Liao, Z., Zhang, S., Li, K., Zhang, G., Habetler, T.G. (2019). A survey of methods for monitoring and detecting thermal runaway of lithium-ion batteries. *Journal of Power Sources*, *436*, 226879, https://doi.org/10.1016/j.jpowsour.2019.226879

Lu, L., Han, X., Li, J., Hua, J., Ouyang, M. (2013). A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, *226*, 272–288, https://doi.org/10.1016/j.jpowsour.2012.10.060

Ma, Z., Huo, Q., Wang, W., Zhang, T. (2023). Voltage-temperature aware thermal runaway alarming framework for electric vehicles via deep learning with attention mechanism in time-frequency domain. *Energy*, *278*, 127747, https://doi.org/10.1016/j.energy.2023.127747

Mateev, V., Marinova, I., Kartunov, Z. (2019). Gas leakage source detection for Li-ion batteries by distributed sensor array. *Sensors*, *19*(13), 2900, https://doi.org/10.3390/s19132900

Maxim Integrated (n.d.). *DS18B20 programmable resolution 1-Wire digital thermometer (data sheet).* Analog Devices data sheet, accessed 30 October 2026.

Nordby, J. (2019). *emlearn: Machine learning inference engine for microcontrollers and embedded devices.* Zenodo software.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830, Retrieved from https://jmlr.org/papers/v12/pedregosa11a.html

Ray, P.P. (2022). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University – Computer and Information Sciences*, *34*(4), 1595–1623, https://doi.org/10.1016/j.jksuci.2021.11.019

Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., ... Dormann, C.F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929, https://doi.org/10.1111/ecog.02881

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437, https://doi.org/10.1016/j.ipm.2009.03.002

Sun, P., Bisschop, R., Niu, H., Huang, X. (2020). A review of battery fires in electric vehicles. *Fire Technology*, *56*(4), 1361–1410, https://doi.org/10.1007/s10694-019-00944-3

Texas Instruments (2015). *INA219 zerø-drift, bidirectional current/power monitor with I^2C interface (data sheet SBOS448G).* Texas Instruments. Retrieved from https://www.ti.com/lit/ds/symlink/ina219.pdf

Wang, Q., Ping, P., Zhao, X., Chu, G., Sun, J., Chen, C. (2012). Thermal runaway caused fire and explosion of lithium ion battery. *Journal of Power Sources*, *208*, 210–224, https://doi.org/10.1016/j.jpowsour.2012.02.038

Weng, Y., Guan, W., Ababei, C. (2024). Prediction of remaining useful life and cell temperature for Li-ion batteries using TinyML. *2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS)* (pp. 562–566). IEEE.

Zhang, X., Pan, Y., Cao, Y., Liu, B., Yu, X. (2024). Cloud-based battery failure prediction and early warning using multi-source signals and machine learning. *Journal*

of Energy Storage, *93*, 112004, https://doi.org/10.1016/j.est.2024.112004

Zhao, Y., Liu, P., Wang, Z., Zhang, L., Hong, J. (2017). Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods. *Applied Energy*, *207*, 354–362, https://doi.org/10.1016/j.apenergy.2017.05.139