Yashi Jha, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Solar Panel Battery Level Measurement System: Hardware Implementation, Measurements and Instrumentation, and Machine Learning-Based Battery Degradation Analysis

Yashi Jha[1], Dheeksha M[2], Naman Sudhir[3], Adit Halder[4], Dave Kurian[5], Assistant Professor Dr. Thamimaran A[6]

Department of Electrical and Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India[1,2,3,4,5,6]

Abstract- This paper presents a low-cost, solar-charged battery monitoring system developed as a Measurements and Instrumentation course project. A 6 V, 2 W photovoltaic panel charges an 18650 lithium-ion cell through a TP4056-based constant-current/constant-voltage charging module. The prototype has two sensing channels: an INA219 current/power monitor on the I2C bus, which measures bus voltage, charging current and power, and a resistive divider (R1 = 10 kΩ, R2 = 1.5 kΩ) into the 10-bit ADC of an Arduino UNO R3. The firmware reads the INA219 once per second, shows voltage, current and power on a 16 × 2 I2C LCD and the serial monitor, and drives red, yellow and green charging-status LEDs. An instrumentation analysis compares the two voltage channels. The divider resolves about 37.5 mV per ADC step at the battery and needs calibration to meet a ±0.05 V target, whereas the INA219 resolves 4 mV with a datasheet maximum bus-voltage error of ±0.5 %, about ±21 mV at 4.2 V. The hardware and software are linked through the recorded measurements: voltage, current and power readings logged from the prototype were used as the basis for generating a larger battery-ageing dataset, on which machine-learning models were trained to predict the state R2 of 0.9906 and an RMSE of 0.631 %, and predicts that 80 % end of life is reached at 929 cycles. An XGBoost regressor using depth of discharge, charge rate, temperature and cycle count achieves a test RMSE of 2.107 % and an MAE of 1.485 % on unseen cells. The paper also documents the error budget, a calibration procedure against a digital multimeter, and the limitations of the prototype.

Keywords- Battery monitoring, current and power measurement, INA219, voltage divider, analog-to-digital conversion, Arduino, state of health, exponential regression, XGBoost.

I. INTRODUCTION

Background

Small photovoltaic (PV) systems commonly store energy in lithium-ion cells, which offer high en-ergy density and low self-discharge. The useful-ness of such a system depends on knowing two quantities. The state of charge (SOC) is the charge currently available relative to the charge at full capacity, and it tells the user how much en-ergy is left [3]. The state of health (SOH) is the present capacity relative to the rated capacity, and it falls as the cell ages through mechanisms such as solid-electrolyte-interphase (SEI) growth and lithium plating [1], [2]. A cell is conventionally treated as having reached end of life (EOL) when its SOH falls to 80 %.

II. PROBLEM STATEMENT

Many low-cost solar charging setups provide no indication of battery condition. The user cannot tell when the cell is close to empty or when it has degraded, and laboratory battery analysers or dedicated fuel-gauge ICs add cost and complexity.

The problem addressed here is to build an in-expensive instrument that measures battery voltage with known and controllable error, dis-plays voltage, charging current and power in real time, and to complement it with a reproducible method for estimating the SOH of the cell from the recorded measurements.

© 2026 Yashi Jha. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

III. MOTIVATION

The project applies core Measurements and In-strumentation concepts (signal conditioning, ana-log-to-digital conversion, quantisation, calibration and error analysis) to a practical device. Data-driven methods are increasingly used to predict battery life [6], and linking such methods to the readings of a simple voltage and current monitor extends the instrument from measuring the present state of the cell to estimating its health.

Objectives

- Build the energy path: solar panel → TP4056 charging module → 18650 cell.
- Measure battery voltage, charging current and power with an INA219 sensor, provide an ana-log voltage channel through a resistive divider into the Arduino 10-bit ADC, and display the readings on a 16×2 LCD with LED charging-status indication.
- Quantify the measurement uncertainty of both voltage channels and define a calibration procedure against a digital multimeter (DMM), with a target voltage accuracy of ±0.05 V.
- Use the readings recorded from the hardware to generate a larger dataset, develop exponential-regression and XGBoost SOH models on it, compare the exponential model with a linear baseline, and evaluate both on held-out data.

Scope of the Project

The work covers a single 18650 cell. Voltage, current and power are measured; temperature is not measured by the hardware. The current firm-ware does not compute SOC: it reports the measured quantities and a three-state charging indication. Because the prototype cannot run multi-month ageing tests, the readings recorded from the hardware are expanded into a larger generated dataset, and the SOH models are trained and evaluated offline on a PC using this dataset (see Hardware–Software Integration).

IV. LITERATURE REVIEW

Existing Research- Vetter et al. [1] reviewed lithium-ion ageing and identified SEI growth, loss of active material and lithium plating as the principal causes of capacity fade. Waldmann et al. [2] showed experimentally that ageing has a temperature minimum near 25 °C: below it, lithium plating accelerates age-ing, while above it, cathode degradation and SEI growth dominate. Hannan et al. [3] reviewed SOC estimation methods, from direct voltage-based approaches to coulomb counting, model-based filters and data-driven methods. Ng et al. [4] pro-posed an enhanced coulomb-counting method that also evaluates SOH from the maximum re-leasable capacity. Goud et al. [5] proposed an online method of estimating the SOH of a Li-ion battery that accounts for temperature, charge/discharge rate and depth of discharge. Severson et al. [6] cycled 124 commercial cells and used machine learning on early-cycle data to predict cycle life with 9.1 % test error. Chen and Guestrin [7] introduced XGBoost, a regularised, scalable gradient tree-boosting system widely used for tabular regression.

V. REVIEW OF EXISTING METHODS

Table 1 summarises the main families of bat-tery-state estimation methods and their suitability for a low-cost voltage-and-current instrument.

Table 1: Battery-state estimation methods

Method	Principle	Strength	Limitation
Voltage / OCV mapping [3]	Map terminal voltage to SOC	Simple; needs only voltage	Approximate; load and temperature dependent
Coulomb counting [4]	Integrate current over time	Accurate over short periods	Needs current sensor; drift
Model-based [3], [9]	Equivalent circuit + observer	Robust, dynamic	Needs cell parameters, current
Data-driven ML [6], [7]	Learn from ageing data	Captu res nonlin ear effects	Needs large, good-quality data

Limitations of Existing Systems

Hobby voltmeter modules are rarely calibrated and do not state their error. Fuel-gauge ICs and full battery management systems need a current-sense resistor and cell-specific characterisation. The data-driven studies in [6] rely on long labor-atory cycling campaigns with specialised equip-ment, which is outside the reach of a course pro-ject.

Research Gap

Few low-cost educational systems combine a documented instrumentation error analysis with an SOH modelling workflow. This project ad-dresses that gap at a modest scale: a calibrated voltage and current instrument with an explicit er-ror budget, whose recorded readings are used to build the dataset for a reproducible SOH model-ling pipeline.

VI. METHODOLOGY

Proposed Solution

The solution has two connected parts: (i) a hard-ware instrument that charges the cell from the panel and measures and displays its voltage, charging current and power, and (ii) a software at its IN− pin with a 4 mV step, using a 12-bit ADC [13]. Current and power follow as (4)

The device computes the power product intern-ally once calibrated [13], and the firmware reads Vbus, I and P directly. For a 0.1 Ω shunt, a com-mon value on INA219 breakout boards, the 10 µV shunt step corresponds to 0.1 mA.

Charging-status logic. Once per second the firm-ware lights one LED: red if I < 20 mA (little or no charging current), otherwise green if study that uses the readings recorded from the hardware to generate a larger dataset and estimVbus≥ 4.15 V (near full charge), otherwise yellow ates SOH with exponential regression and XG-Boost.

Methodology Overview

The work proceeded as follows: design and as-sembly of the energy and measurement paths; firmware for sampling, conversion and display; error analysis and calibration of both measure-ment channels; recording of readings from the prototype; generation of a larger ageing dataset from the recorded readings; fitting and evaluation of the SOH models; and comparison against a linear baseline and an 80 % EOL classification.

Algorithms and Techniques

Voltage measurement. With R1 between battery positive and the measurement node and R2 between the node and ground, the divider output is

$$V_{out} = V_{bat}\frac{R_2}{R_1 + R_2} \quad (1)$$

Assuming a nominal 5.0 V ADC reference, a 10-bit reading N (0–1023) is converted by

$$V_{out} = \frac{N \times 5.0}{1023} \quad (2)$$

and the battery voltage is recovered by inverting (1):

$$V_{bat} = V_{out}\frac{R_1 + R_2}{R_2} \quad (3)$$

For R_1 = 10 kΩ and R_2 = 1.5 kΩ the multiplier $(R_1+R_2)/R2$ is 7.667, so one ADC step (4.89 mV at A0) corresponds to about 37.5 mV at the battery.

A full cell at 4.2 V gives V ≈ 0.548 V, well with- (charging). The conditions are evaluated in that order.

Calibration and error. A channel reading Vm (INA219 or A0 channel) is compared with a DMM reading Vdmm taken at the same time. The abso-lute error and absolute percentage error (APE) are

$$E = V_m - V_{dmm}, \quad \text{APE} = \frac{|E|}{V_{dmm}} \times 100\% \quad (5)$$

If a systematic error is found, a linear correction Vcal = m Vm + c, fitted by least squares to the paired readings, or an adjusted reference constant in (2), is applied in firmware. The INA219 current is checked in the same way against a DMM connected in series. Exponential regression (Model 1). Capacity fade with cycle count n is modelled as

$$\text{SOH}(n) = a\,e^{-bn} \quad (6)$$

where a is the initial SOH (%) and b is the fade rate per cycle. Initial values are obtained from the log-linear fit ln SOH = ln a − bn, and the paramet-ers are then refined by nonlinear least squares (SciPy c u r v e _ f i t). The predicted cycles to EOL follow from setting SOH = 80 %:

$$n_{EOL} = \frac{\ln(a/80)}{b} \qquad (7)$$

A straight-line fit SOH = c − mn on the same training data serves as the baseline.

XGBoost regression (Model 2). XGBoost predicts SOH as a sum of *K* regression trees, $\hat{y} = \Sigma k\, fk(\mathbf{x})$, each tree fitted to the residual gradient of a regularised squared-error objective [7]. The input vector x contains depth of discharge (DoD, %),charge rate (C-rate), temperature (°C) and cycle count.

Evaluation. Regression accuracy on held-out data is measured by the coefficient of determination R2, the root-mean-square error (RMSE) and the mean absolute error (MAE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \qquad (8)$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \qquad (9)$$

Workflow

The link between the hardware readings and the ML dataset is described under Hardware–Software Integration; the ML data flow and the firmware loop are detailed under Data Flow Diagram and Implementation Details. Both ML scripts use a fixed random seed (42), so the generated data, the train/test split and the results are reproducible.

System Architecture and Design

System Architecture

The system has three paths (Figure 1). The en-ergy path converts sunlight into stored charge. The measurement path senses battery voltage and charging current through two channels, pro-cesses them, and presents the results on the LCD, the serial monitor and the status LEDs. The software path runs offline on a PC: it takes the readings logged from the measurement path, generates a larger dataset from them, and estim-ates the SOH of the cell.

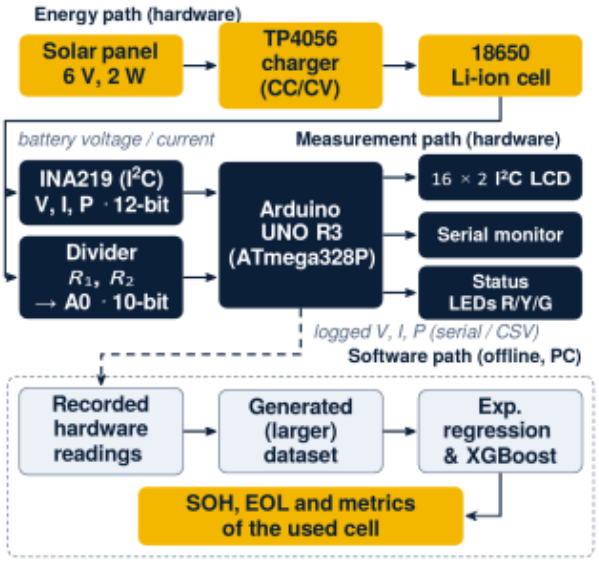


Figure 1: System block diagram showing the energy path, the measurement path and the software path, which uses the readings recorded from the hardware.

Block Diagram

In the energy path, the panel output feeds the TP4056 module input (IN+/IN−) and the module charges the cell through B+/B−.

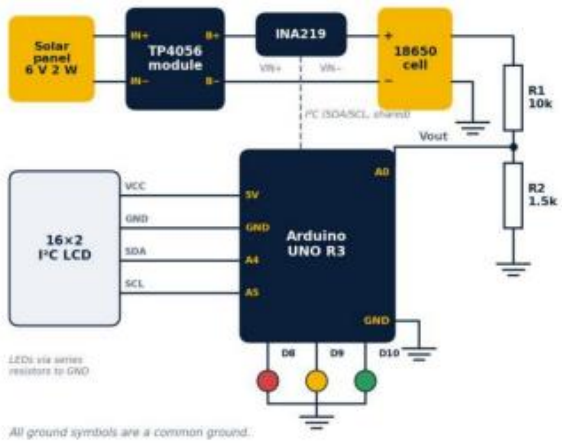


Figure 2: Circuit and wiring diagram: charging path with the INA219, divider into A0, shared I2C bus for the LCD and INA219, and status LEDs.

The INA219 shunt is in series with the B+ connection; the firmware treats its bus voltage as the battery voltage and its current as the charging current, which holds only for this placement. The INA219 shares the I2C bus (SDA on A4, SCL on A5 of the UNO R3) with the LCD. The R1–R2 divider across the cell feeds A0, and the status LEDs are driven from D8 (red), D9 (yellow) and D10 (green). All grounds are common. The wiring is shown in Figure 2.

System Components

Table 2 lists the hardware components. The nom-inal panel current at its rated maximum-power point is I = P/V = 2 W/6 V ≈ 0.33 A. Actual output varies with irradiance, angle and temperature, and the open-circuit voltage can exceed 6 V, so the charger input range must be checked against the measured panel voltage.

Table 2: Hardware components

Component	Key specification	Role
Solar panel	6 V, 2 W rated	Energy source
TP4056 module	Single-cell Li-ion CC/CV charger; 4.2 V termination [10]	Charge control
18650 Li-ion cell	≈3.6–3.7 V nominal, 4.2 V full charge	Energy storage
INA219 module	12-bit; bus 4 mV/LSB, shunt 10 µV/LSB; I²C [13]	Voltage, current, power
Resistor divider	R_1 = 10 kΩ, R_2 = 1.5 kΩ	Analog voltage channel
Arduino UNO R3	ATmega328P, 10-bit ADC, 5 V logic [11], [12]	Acquisition and processing
16×2 LCD + I²C backpack	2×16 characters; address 0x27	Display of V, I, P
LEDs (red, yellow, green)	Driven from D8, D9, D10	Charging status

Data Flow Diagram

Figure 3 shows the data flow of the ML study. The readings recorded from the hardware are used to generate the larger dataset (see Hardware–Software Integration); a measured CSV file can also be passed on the command line. Each script then splits the data, fits the model, and produces metrics, plots and (for Model 2) a saved model file.

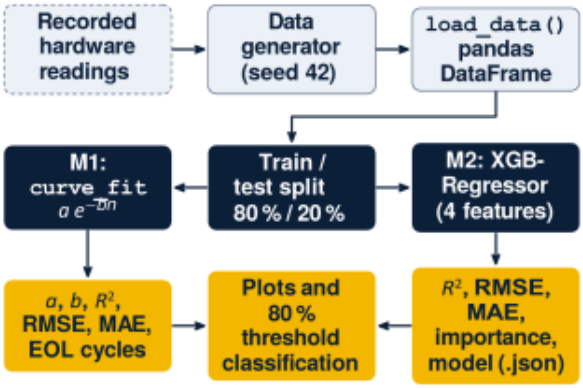


Figure 3: Data flow of the SOH modelling study: recorded hardware readings are expanded into a larger dataset that is used by Model 1 (exponential regression) and Model 2 (XGBoost).

Module Description

Energy module. The panel supplies the TP4056 module, which charges the cell with a constant current followed by a constant 4.2 V phase as the current tapers [10]. The panel must never be con-nected directly to the cell.

Sensing Module

The INA219 senses the charging current through its shunt and the bus voltage at its IN− pin. The R1–R2 divider scales 0–4.2 V to 0–0.548 V for A0 and draws about 4.2 V/11.5 kΩ ≈ 0.37 mA continuously at 4.2 V.

Processing Module

The ATmega328P reads V , I and P from the INA219 over I2C once per second, applies the LED logic and updates the outputs.

Display and indication module. The 16×2 LCD at I2C address 0x27 shows voltage and current on the first row and power on the second. The same values are printed to the serial monitor, and one of three LEDs indicates the charging state.

ML analysis module. Two Python scripts imple-ment Model 1 and Model 2 on the dataset gener-ated from the recorded hardware readings.

Hardware–Software Integration

The hardware and the software parts of the pro-ject are connected through the measurements, as shown in Figure 4.

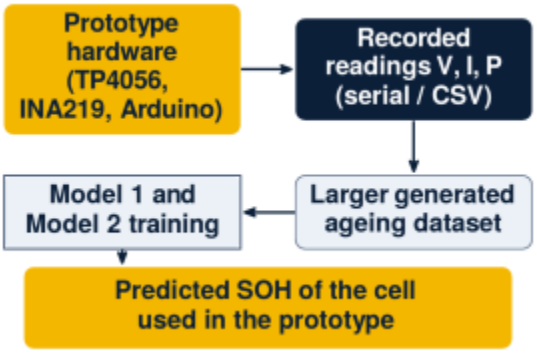


Figure 4: Hardware–software integration: readings recor-ded from the prototype are expanded into a larger dataset, which trains the models used to predict the SOH of the cell.

Recording the hardware readings. While the cell was being charged from the panel, the prototype measured the battery voltage, charging current and power once per second with the INA219, and these readings were printed to the serial monitor and recorded on the PC. The readings were also cross-checked against the DMM following the procedure described under Testing Procedure.

Generating a larger dataset. A single cell in a course project can only be observed for a limited number of charge cycles, which is far too little data to train or test an SOH model. The recorded readings were therefore used as the basis for generating a larger battery-ageing dataset: the data generator in each script produces many cycle points (Model 1) and many cells cycled un-der different conditions (Model 2), following the capacity-fade behaviour reported in the literature [1], [2]. The generated dataset is summarised in Table 3.

Training and prediction. Both models are trained and tested on the generated dataset. Once trained, the models are used to predict the SOH of the cell used in the prototype: Model 1 from its cycle count, and Model 2 from its operating con-ditions (depth of discharge, charge rate, temper-ature and cycle count). Both scripts also accept a CSV file of measured data, so further readings recorded from the hardware can be added and the models retrained without changing the code.

Implementation

Technologies and Tools Used

The firmware is written in C/C++ in the Arduino IDE using the Wire, LiquidCrystal_I2C and Adafruit_INA219 libraries. The ML study uses Python 3 with NumPy, pandas, SciPy (c u r v e _ f i t), scikit-learn [8] for splitting and metrics, XGBoost [7] and Matplotlib.

Hardware Requirements

- 6 V, 2 W solar panel; TP4056-based charging module; 18650 Li-ion cell in a holder.
- Arduino UNO R3 with USB cable; INA219 current/power sensor module; 10 kΩ and
- 1.5 kΩ resistors; 16×2 LCD with I2C backpack;
- red, yellow and green LEDs with series resist-ors.
- Breadboard and jumper wires; digital multimeter for calibration.
- For standalone operation, a suitably rated 5 V boost converter, since a single cell cannot supply a regulated 5 V.

Software Requirements

- Arduino IDE with the LiquidCrystal_I2C and Adafruit_INA219 libraries.
- Python 3 with numpy, pandas, scipy, scikit-learn, xgboost and matplotlib.

Dataset Description

Both models are trained on a dataset generated from the readings recorded with the prototype, because a single cell cannot be cycled long enough within the project to observe its full capa-city fade. The generator parameters represent typical 18650 behaviour. Table 3 summarises both datasets.

Table 3: Generated datasets

Item	Model 1	Model 2
Generator	SOH = 100 $e^{-0.00024n}$ + noise	SOH = 100 e^{-bn} + noise, b = 0.00018 × stress
Samples	400 points	120 cells × 25 cycle points = 3000 rows

Inputs	Cycle count (0–1000)	DoD 20–100 %, C-rate 0.2–2.0, temperature 5–55 °C, cycle count 0–1200
Noise (σ)	0.6 % SOH	0.5 % SOH
Split	Random 80/20: 320 train, 80 test	By cell 80/20: 96 cells (2400 rows) train, 24 cells (600 rows) test

In Model 2, the stress factor rises with deeper discharge, (DoD/80)1.3, faster charging, 1 + 0.35(C − 0.5), and temperatures away from 25 °C (an extra 4 % per °C above 25 °C and 2 % per °C below 15 °C). This mimics the qualitative trends reported in [1], [2]. Splitting by cell (G r o u p S h u f f l e S p l i t) ensures that every test cell is unseen during training.

Implementation Details

Hardware. The circuit of Figure 2 was assembled with the Arduino powered over USB during test-ing, while charging and sensing were tested sep-arately; this avoids connecting the cell directly to the 5 V or VIN pin.

Firmware. The sketch follows Figure 5. s e t u p ( ) starts the serial port at 9600 baud and the I2C bus, configures D8–D10 as outputs, initialises the LCD and shows a start-up message. If the INA219 is not detected, the LCD shows "INA219 Error!" and the red LED blinks indefinitely. Otherwise, l o o p ( ) runs once per second: it reads bus voltage, current and power, prints them to the serial monitor, updates the LEDs, and writes "V:" and "I:" on the first LCD row and "Power:" in mW on the second. The serial output of this loop is the source of the recorded readings used for the generated dataset. The current firmware does not read A0; the analog channel is characterised analytically under Results and Discussion.

Model 1. The script fits (6) to the training points, prints the parameters with standard errors, test R2, RMSE, MAE and n, plots the fit with the lin-EOLear baseline, and reports an example prediction at 500 cycles. Model 2. X G B R e g r e s s o r is configured with 600 es-timators, learning rate 0.05, maximum depth 4, row subsampling 0.8, L2 regularisation λ = 1 and early stopping after 40 rounds without improve-ment. The script prints test metrics and feature importance, saves a predicted-versus-actual plot and the model (s o h _ x g b _ m o d e l . j s o n), and then accepts user inputs for an SOH prediction.

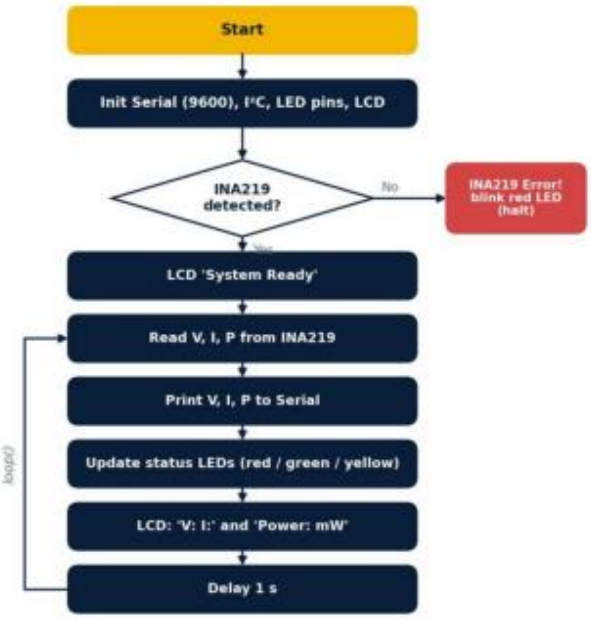


Figure 5: Firmware flowchart: INA219 measurement, serial output, LED indication and LCD display.

Experimental Setup

Experimental Environment

Hardware tests use the assembled prototype, a USB-powered Arduino and a DMM as the refer-ence instrument, with the panel exposed to avail-able sunlight for charging tests. The ML experi-ments run on a PC with the Python stack listed under Software Requirements.

Testing Procedure

- Measure the cell voltage with the DMM and re-cord the INA219 reading at the same moment; measure the charging current with the DMM in series and compare it with the INA219 current.
- Repeat at several safe voltage levels within the cell's limits, without overcharging or over-dis-charging.
- Compute E and APE using (5); if a systematic error appears, fit and apply a calibration cor-rection, then re-test.

- Record the voltage, current and power read-ings from the serial monitor and use them to generate the larger dataset.
- Run both ML scripts with seed 42, evaluate on held-out data, and classify each test sample as Degraded (< 80 %) or Healthy (≥ 80 %) using actual and predicted SOH.

Evaluation Metrics

Hardware accuracy is assessed with E and APE from (5). Regression models are assessed with R2, RMSE and MAE from (8) and (9). For the 80 % threshold classification, with TP, FP and FN coun-ted for the chosen positive class,

$$\mathrm{Pr} = \frac{TP}{TP + FP}, \quad \mathrm{Re} = \frac{TP}{TP + FN} \tag{10}$$

$$F_1 = \frac{2\,\mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{11}$$

Baseline Methods

A linear regression of SOH on cycle count, fitted on the same training split, is the baseline for Model 1. No separate baseline was implemented for Model 2.

VII. RESULTS AND DISCUSSION

Measurement Design Values

For the A0 channel, Table 4 lists the theoretical divider output and ideal ADC count across the us-able cell voltage range, computed from (1) and (2) with nominal component values. These are design values, not measurements.

Table 4: Theoretical divider output and ADC count

V_{bat} (V)	V_{out} (V)	Ideal ADC count
3.0	0.391	80
3.5	0.457	93
3.7	0.483	99
4.0	0.522	107
4.2	0.548	112

Error Budget

Table 5 compares the error sources of the two voltage channels at Vbat = 4.2 V. Uncalibrated, the A0 channel can exceed the ±0.05 V target through resistor tolerance or reference error alone, whereas the INA219 datasheet maximum error lies within the target. The A0 channel there-fore requires DMM calibration, and the INA219 should still be verified against the DMM.

Table 5: Voltage measurement error budget at 4.2 V

Channel	Source	Error at battery
A0 divider	Quantisation, ±0.5 LSB (37.5 mV/LSB)	±19 mV
A0 divider	Resistor tolerance ±1 % each, worst case	≈ ±73 mV
A0 divider	ADC reference, per 1 % deviation	±42 mV
INA219	Quantisation, ±0.5 LSB (4 mV/LSB)	±2 mV
INA219	Bus-voltage error, ±0.5 % max [13]	±21 mV

Hardware Readings

The voltage, charging current and power read-ings were recorded from the serial monitor during charging and checked against the DMM using the procedure described under Testing Procedure. These recorded readings form the basis of the generated dataset used by both models.

Regression Results

Model 1. The fitted parameters are a = 99.935 ±0.074 % and b = 0.000239 ± 0.000001 per cycle, close to the generator values (100, 0.00024). Table 6 compares Model 1 with the linear baseline, and Chart 1 shows the fit.

Table 6: Regression results on held-out test data

Model	R^2	RMSE (%)	MAE (%)	EOL (cycles)
Linear baseline	0.9890	0.685	0.568	916
Exponential (M1)	0.9906	0.631	0.512	929
XGBoost (M2)	≈0.958*	2.107	1.485	—

Model 1 and baseline: 80 test points. Model 2: 600 test rows from 24 unseen cells. *Computed from the reported RMSE and the variance of the Model 2 test set (R2 = 1 − RMSE2/σ2).

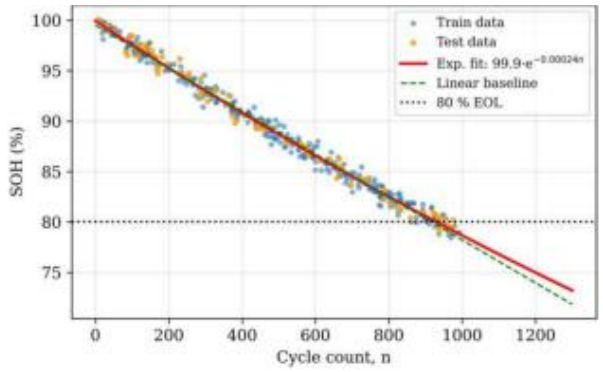


Chart 1: SOH data with the fitted exponential model, the lin-ear baseline and the 80 % EOL threshold (Model 1).

Model 2. On the 24 unseen test cells, XGBoost achieved an RMSE of 2.107 % and an MAE of 1.485 %. The feature importances (Chart 2) are cycle count 0.438, depth of discharge 0.371, temperature 0.135 and charge rate 0.056.

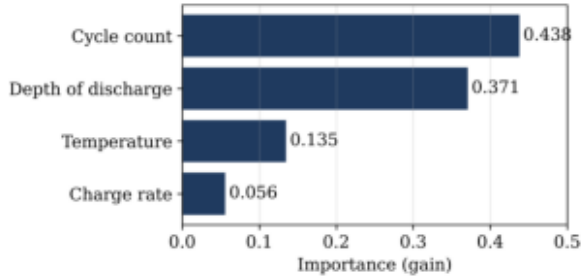


Chart 2: XGBoost feature importance for SOH prediction (Model 2).

80 % Threshold Classification

Chart 3 shows the confusion matrices obtained by thresholding actual and predicted SOH at 80 %. Table 7 reports the metrics with Degraded as the positive class, the class of practical in-terest, and Healthy-class values for reference.

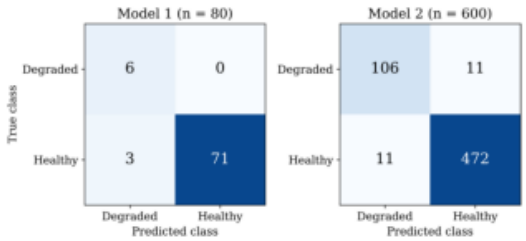


Chart 3: Confusion matrices at the 80 % SOH threshold: Model 1 (left, n = 80) and Model 2 (right, n = 600).

Table 7: Classification at the 80 % SOH threshold

Metric	Model 1 (*n* = 80)	Model 2 (*n* = 600)
Accuracy	96.25 %	96.33 %
Precision (Degraded)	0.667	0.906
Recall (Degraded)	1.000	0.906
F1 (Degraded)	0.800	0.906
Precision / Recall / F1 (Healthy)	1.000 / 0.959 / 0.979	0.977 / 0.977 / 0.977

Tables and Comparisons

Tables 4 and 5 characterise the measurement chain. Table 6 compares the regression models, and Table 7 compares the threshold classification of both models.

Performance Analysis

For voltage measurement, the INA219 channel of-fers about nine times finer resolution than the A0 channel (4 mV vs. 37.5 mV) and a datasheet error bound within the ±0.05 V target. It also adds cur-rent and power, which the A0 channel cannot provide.

Model 1 improves on the linear baseline in every metric, but only slightly (R2 0.9906 vs. 0.9890). Model 2 has larger errors (RMSE 2.107 %) be-cause it is evaluated on unseen cells whose SOH depends on four interacting factors and ranges down to about 50 %, a far harder task than fitting one fade curve. In classification, Model 1 caught all 6 degraded test samples but raised 3 false alarms, giving a Degraded-class precision of 0.667. Model 2 identified 106 of 117 degraded samples with 11 false alarms.

Discussion of Findings

The close agreement between the exponential and linear fits within 0–1000 cycles is expected: bn ≤ 0.24, so e−bn is nearly linear over the ob-served range. The difference appears in extra-polation, with EOL predicted at 929 cycles by the exponential model and 916 cycles by the linear baseline. The recovered parameters match the generator closely, which confirms that the fitting procedure works.

The Model 2 importance ranking mirrors the generator: cycle count and DoD (which enters with exponent 1.3) dominate, the asymmetric temperature term is third, and charge rate (a mild linear factor) is last. These results show that the pipeline recovers known structure from the data-set generated from the hardware readings. Long-term ageing of real cells would be needed to confirm the predicted fade over hundreds of cycles.

Two caveats apply to Model 2. First, the test set is also used for early stopping, which makes the reported test metrics slightly optimistic. Second, results vary slightly with library versions; for example, re-running the unchanged script with XGBoost 3.4.1 gave an RMSE of 1.960 %. The two classification test sets also differ in size and degraded-class count (6 vs. 117), so the models cannot be compared directly on these metrics.

VIII. CONCLUSION

Summary of the Project

A solar-charged 18650 battery monitor was designed using a TP4056 charger, an INA219 current/power sensor and a resistive-divider channel into an Arduino UNO, with a 16×2 I2C LCD, serial output and three charging-status LEDs. The readings recorded from the prototype were used to generate a larger dataset, on which SOH degradation was modelled with exponential regression and XGBoost to predict the SOH of the cell used.

Key Findings

- The INA219 resolves 4 mV with a datasheet maximum error of about ±21 mV at 4.2 V, within the ±0.05 V target, and adds current and power measurement.
- The A0 divider channel resolves about 37.5 mV per step and uses about 11 % of the ADC range; its resistor-tolerance and reference er-rors can each exceed ±0.05 V, so it requires calibration.
- The exponential model achieved R2 = 0.9906 and RMSE = 0.631 %, and predicted EOL at 929 cycles.
- XGBoost achieved RMSE = 2.107 % and MAE = 1.485 % on unseen cells, with cycle count and DoD the most important inputs.

Objectives Achieved

All four objectives were achieved: the energy path was built, the measurement and display sys-tem was implemented, the error budget and cal-ibration procedure were completed, and the SOH models were developed on the dataset generated from the recorded hardware readings.

Limitations

- No SOC algorithm is implemented in the firmware, and temperature is not measured.
- The A0 divider channel is wired but not read by the current firmware.
- The INA219 measures bus voltage at its IN− pin, so the reading equals the battery voltage only if the sensor is directly in series with the cell.
- The low-current (red) condition is checked first, so a fully charged cell whose charging current has tapered may show red rather than green.
- The prototype cell could not be aged over hundreds of cycles within the project, so the long-term fade predicted by the models has not yet been confirmed on the real cell.
- Model 2 uses the test set for early stopping, and its interactive input reads integers, so fractional charge rates cannot be entered.

Future Work

Possible improvements. Check the full-voltage condition before the low-current condition in the LED logic, so that a full cell shows green. Read A0 in firmware and display it alongside the INA219 value as a cross-check. If the A0 channel is kept, use the ATmega328P internal 1.1 V reference [11] with a one-time calibration; the step size falls to about 8.2 mV at the battery. Hold out a separate validation set for early stopping, and read charge rate and temperature as floating-point values.

Future enhancements. Integrate the INA219 cur-rent over time (coulomb counting [4]) to estimate SOC, and add a temperature sensor. Log voltage, current and power to an SD card or an IoT dash-board. Record readings from the cell over many more cycles and pass them to the existing CSV input of both scripts to retrain and validate the models. Use a solar-compatible charging stage and a boost converter for standalone operation.

Potential applications. Monitoring small sol-ar-charged batteries, laboratory demonstrations of PV charging, signal conditioning and ADC op-eration, and a foundation for low-cost battery data logging and health monitoring.

Acknowledgements

The authors thank Dr. Thamimaran A, Assistant Professor (Senior), and the Department of Elec-trical and Electronics Engineering, Vellore Insti-tute of Technology, Vellore, for guidance and support.

REFERENCES

1. Vetter, J. et al. (2005). Ageing mechanisms in lithium-ion batteries. Journal of Power Sources, 147(1–2), 269–281. doi:10.1016/j.jpowsour.2005.01.006
2. Waldmann, T., Wilka, M., Kasper, M., Fleischhammer, M., & Wohlfahrt-Mehrens, M. (2014). Temperature dependent ageing mechanisms in lithium-ion batteries – A post-mortem study. Journal of Power Sources, 262, 129–135. doi:10.1016/j.jpowsour.2014.03.112
3. Hannan, M. A., Lipu, M. S. H., Hussain, A., & Mohamed, A. (2017). A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. Renewable and Sustainable Energy Reviews, 78, 834–854. doi:10.1016/j.rser.2017.05.001
4. Ng, K. S., Moo, C.-S., Chen, Y.-P., & Hsieh, Y.-C. (2009). Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. Applied Energy, 86(9), 1506–1511. doi:10.1016/j.apenergy.2008.11.021
5. Goud, J. S., Kalpana, R., & Singh, B. (2021). An online method of estimating state of health of a Li-ion battery. IEEE Transactions on Energy Conversion, 36(1), 111–119. doi:10.1109/TEC.2020.3008937
6. Severson, K. A. et al. (2019). Data-driven prediction of battery cycle life before capacity degradation. Nature Energy, 4(5), 383–391. doi:10.1038/s41560-019-0356-8
7. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. doi:10.1145/2939672.2939785
8. Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
9. Plett, G. L. (2015). Battery Management Systems, Volume I: Battery Modeling. Artech House, Norwood, MA, USA.
10. NanJing Top Power ASIC Corp. TP4056: 1A standalone linear Li-ion battery charger with thermal regulation in SOP-8. Datasheet.
11. Microchip Technology Inc. (2020). AT mega48A/PA/88A/PA/168A/PA/328/P megaAVR data sheet, DS40002061B.
12. Arduino. Arduino UNO R3. Product documentation. https://docs.arduino.cc/hardware/uno-rev3/
13. Texas Instruments (2015). INA219 zero-drift, bidirectional current/power monitor with I2C interface. Datasheet SBOS448G.
14. Sharjil Iqbal. "NETWORK ATTACK DETECTION BY GROUP SEARCH OPTIMIZATION USING CONVOLUTIONAL DEEP LEARNING MODEL