

AI-Based EV Battery Health Prediction with Explainable AI and Database Integration

Pranav Fattensing Sable

Student Research Paper

Email: pranavsabale1956@gmail.com | Mobile: 9405806675

Under the guidance of: **Mrs. Gandhi R.S.**

Abstract

The State of Health (SOH) of a lithium-ion battery determines the usable range, safety margin and resale value of an electric vehicle (EV), yet conventional battery management systems estimate it with fixed rules and give the engineer neither an explanation nor a persistent record. This paper proposes an end-to-end framework that combines (i) machine-learning SOH prediction from cycle-level health indicators, (ii) Explainable AI based on SHapley Additive exPlanations (SHAP) to show why the model predicts a given SOH, and (iii) a relational database layer that stores cell metadata, cycle features, predictions, explanations and alerts for querying and retraining. Five regressors (Ridge regression, support-vector regression, a multilayer perceptron, random forest and gradient boosting) are compared under a cell-wise split so that no cell appears in both training and test data. The framework is prototyped on a **physics-inspired synthetic cycling dataset** of 40 cells and 28,201 cycle records (generated because no real cell data was used in this study). On the 12 held-out cells, gradient boosting gave the lowest error (MAE 1.26 percentage points, RMSE 1.70, $R^2 = 0.967$), although the simple Ridge model was close behind. SHAP identified the dV/dQ peak height, internal resistance and cycle count as the dominant drivers, and a SQLite implementation answered the tested queries in 0.02 to 6.2 ms. Because the data are synthetic, the numerical results demonstrate the working of the pipeline and not real-world battery accuracy; validation on public datasets such as NASA and CALCE is identified as the necessary next step.

Index Terms— Electric vehicle, lithium-ion battery, state of health, machine learning, explainable AI, SHAP, database integration, battery management system.

I. INTRODUCTION

Electric vehicles depend on lithium-ion battery packs whose capacity and power capability decline with use. The State of Health (SOH), commonly defined as the ratio of present capacity to rated capacity, is the key indicator of this decline; a pack is usually considered to reach end of life when SOH falls to about 80% [10]. Knowing SOH accurately and early supports safer operation, warranty decisions, second-life reuse and maintenance planning.

SOH cannot be measured directly during driving. Traditional battery management systems (BMS) rely on coulomb counting, look-up tables or equivalent-circuit models, which drift with temperature, ageing and usage. Data-driven methods learn the relation between measurable signals (voltage, temperature, charge time, resistance) and SOH, and have shown strong results on laboratory data [3], [11]. Two practical gaps remain. First, most high-accuracy models are black boxes: an engineer or fleet operator cannot see which signal pushed the estimate down, which limits trust. Second, predictions are rarely stored together with the inputs and explanations that produced them, so the history needed for audit, drift detection and retraining is lost.

This paper addresses both gaps in one design. Gradient-boosted and other regressors predict SOH; SHAP [1], [2] explains each prediction; and a relational database keeps cell metadata, cycle features, predictions, per-prediction SHAP values and alerts. The objective is a reproducible, auditable pipeline rather than a claim of record accuracy. The main contributions are: (1) a layered architecture for explainable SOH prediction with database integration; (2) a cell-wise evaluation protocol of five models with grouped cross-validation; (3) a SHAP analysis at global and local level whose outputs are persisted in SQL; and (4) an honest account of limitations, notably that the prototype uses synthetic data.

II. BACKGROUND AND LITERATURE REVIEW

A. Battery Ageing and State of Health

Capacity fade and resistance growth in lithium-ion cells result from mechanisms such as solid-electrolyte-interphase growth and lithium plating, and are accelerated by high temperature, high C-rate and deep discharge. Reviews of SOH estimation group methods into direct measurement, model-based and data-driven approaches, and note that real applications demand methods that work from limited and noisy signals [10]. Ageing is typically slow at first and may accelerate after a "knee" point, which makes late-life prediction harder than early-life prediction.

B. Data-Driven SOH and Lifetime Prediction

Severson et al. showed that features derived from early-cycle discharge voltage curves can predict cycle life with small error using regularised linear models [3]. Roman et al. presented a machine-learning pipeline for SOH estimation from charging data and compared several model families [11]. Hu et al. surveyed battery lifetime prognostics and stressed the role of health indicators and data quality [15]. Public repositories such as the NASA Ames battery data set [4] and the CALCE data from the University of Maryland [16] are widely used for benchmarking. A recurring finding is that well-chosen health indicators often matter as much as model complexity, and that evaluation must split by cell, not by random row, to avoid optimistic estimates.

C. Explainable AI

Explainable AI aims to make model decisions understandable to humans [13]. SHAP assigns each feature an additive contribution based on Shapley values from cooperative game theory [14] and unifies several earlier explanation methods [1]. For tree ensembles, TreeSHAP computes exact Shapley values efficiently and supports both global importance and local, per-prediction explanations [2]. LIME is a model-agnostic alternative based on local surrogate models [6]. In safety-relevant domains such as EV batteries, such explanations help to check whether a model relies on physically sensible signals.

D. Database Integration

Fleet-scale battery analytics need persistent storage of telemetry, model outputs and alerts. Relational databases provide structured schemas, foreign-key integrity, indexes and declarative queries, and lightweight engines such as SQLite [12] allow a prototype to be built without a server. Storing explanations next to predictions makes it possible to answer questions such as which feature most often triggered recent alerts.

E. Research Gap

The literature offers accurate predictors, explanation methods and storage technologies, but they are usually studied separately. This work contributes an integrated, auditable design that links prediction, explanation and persistence, evaluated with a leakage-free cell-wise protocol.

III. PROBLEM STATEMENT AND OBJECTIVES

The problem is that SOH estimators deployed in EVs are either simple but inaccurate, or accurate but opaque and unrecorded. A model that cannot justify its output is hard to trust, and a prediction that is not stored cannot be audited or used to improve the model.

Objectives:

- Predict SOH from cycle-level health indicators with low error on cells never seen during training.
- Compare classical and ensemble regressors using a cell-wise split and grouped cross-validation.
- Explain predictions globally and locally using SHAP and check that the drivers are physically sensible.
- Design a relational schema that stores cell metadata, features, predictions, explanations and alerts.
- Measure the practicality of the database layer through insertion cost and query latency.
- Report limitations transparently, including the use of synthetic data.

IV. PROPOSED SYSTEM

The system is modular and can be run as one service or as separate components. Its stages are data acquisition, preprocessing and feature extraction, model training and selection, explanation, database storage and application services such as dashboards and alerts. Fig. 1 summarises the closed loop; the detailed architecture is shown in Section V.

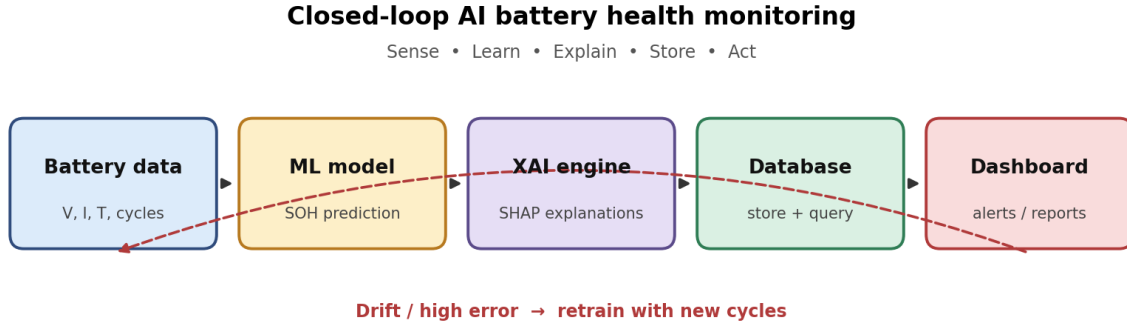


Fig. 1. Conceptual closed loop of the proposed AI-based battery health monitoring system. The red path indicates retraining when drift or high error is detected.

A. Data Acquisition

In deployment, inputs come from BMS sensors (voltage, current, temperature), charger or cyclers logs and vehicle usage context such as ambient temperature. For research, public datasets such as NASA [4] and CALCE [16] can be used. Each record is tied to a cell identifier and a cycle index.

B. Preprocessing and Health-Indicator Extraction

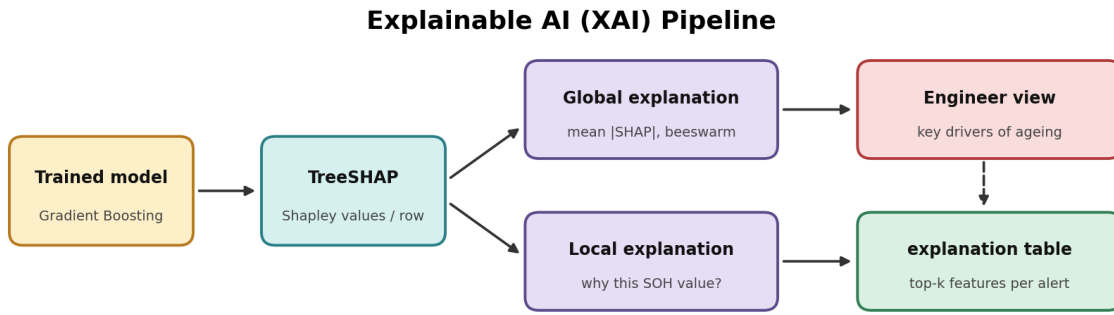
Records are cleaned of missing and out-of-range values, segmented into charge and discharge phases and converted to cycle-level health indicators: discharge time, constant-current charge time, average discharge voltage, internal resistance, maximum and average temperature, the height of the dV/dQ (incremental-capacity-type) peak, cumulative ampere-hour throughput and the cycle index, together with operating conditions (C-rate, depth of discharge, ambient temperature). The target is SOH, the capacity of a cycle divided by the initial capacity.

C. Machine-Learning Models

Five regressors are compared: Ridge regression (regularised linear baseline), support-vector regression with an RBF kernel, a multilayer perceptron with two hidden layers (64 and 32 units), a random forest [8] and gradient boosting [7], all from scikit-learn [9]. Inputs are standardised for the scale-sensitive models. The model with the lowest RMSE on held-out cells is selected for deployment.

D. Explainable AI Module

For the selected tree-based model, TreeSHAP computes one Shapley value per feature per prediction. Mean absolute SHAP values give a global importance ranking; the beeswarm and dependence plots show the direction and shape of each feature's effect; and a per-prediction breakdown explains an individual SOH estimate relative to the average prediction (the baseline). Fig. 2 shows the pipeline.



SHAP value of a feature = its additive contribution ($\pm$ SOH) to the prediction relative to the baseline

Fig. 2. Explainable AI pipeline: TreeSHAP produces global and local explanations that are shown to engineers and stored in the database.

E. Database Layer

A relational schema holds five tables (Fig. 3): *battery* (cell metadata), *cycle_data* (health indicators and measured SOH), *prediction* (model output per cycle), *explanation* (SHAP value per feature per prediction) and *alert* (end-of-life warnings). Foreign keys preserve integrity, and indexes on (*cell_id*, *cycle*) and on the prediction foreign key support fast lookups. The prototype uses SQLite; the same schema transfers to MySQL or PostgreSQL for multi-user deployment.

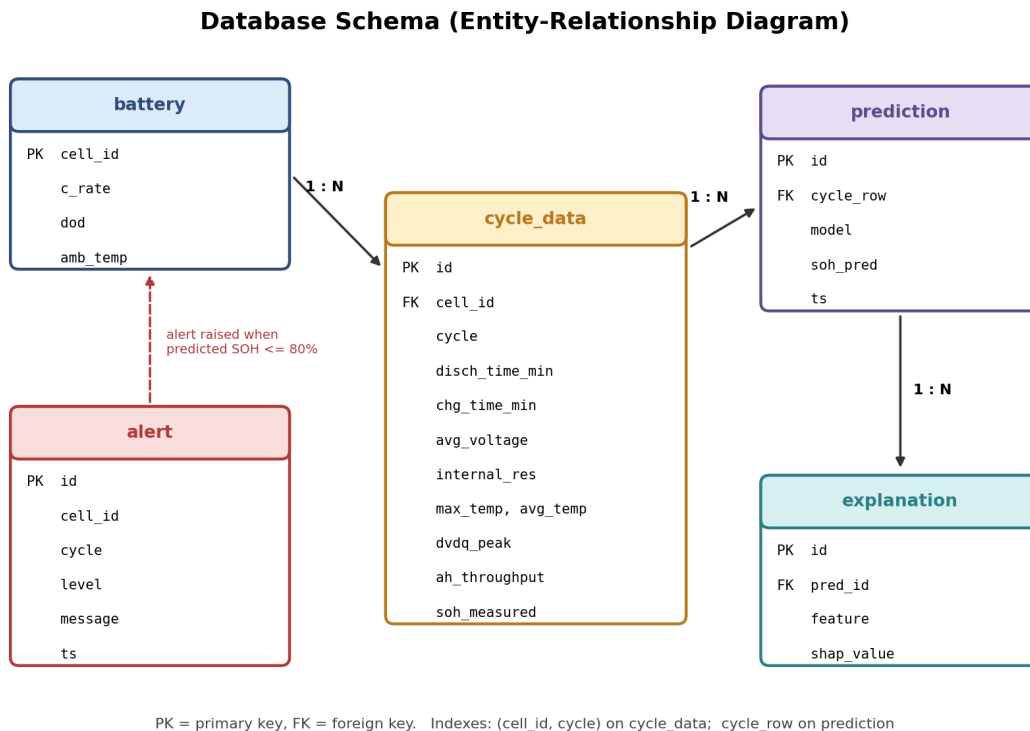


Fig. 3. Entity-relationship diagram of the database schema used for storing cycle data, predictions, SHAP explanations and alerts.

F. Alerts and Application Layer

An alert is raised for a cell when its predicted SOH reaches 80% or below, the common end-of-life criterion [10]. The application layer reads from the database to show SOH trends, the top contributing features for any prediction, active alerts and fleet summaries.

G. Monitoring and Retraining

Stored predictions can later be compared with measured capacity checks. If the error rises or the input distribution shifts (for example, a new battery chemistry or climate), the model is retrained on the accumulated cycles. This loop is described in the architecture but was not exercised in the present prototype.

V. ARCHITECTURE AND WORKFLOW

The architecture separates a data path (sensor data, features, predictions, explanations) from a control path (model selection, alerting, retraining). This separation makes failures traceable: a wrong alert can be followed back to the prediction, the features and finally the raw measurements stored in the database.

Proposed System Architecture: AI-Based EV Battery Health Prediction

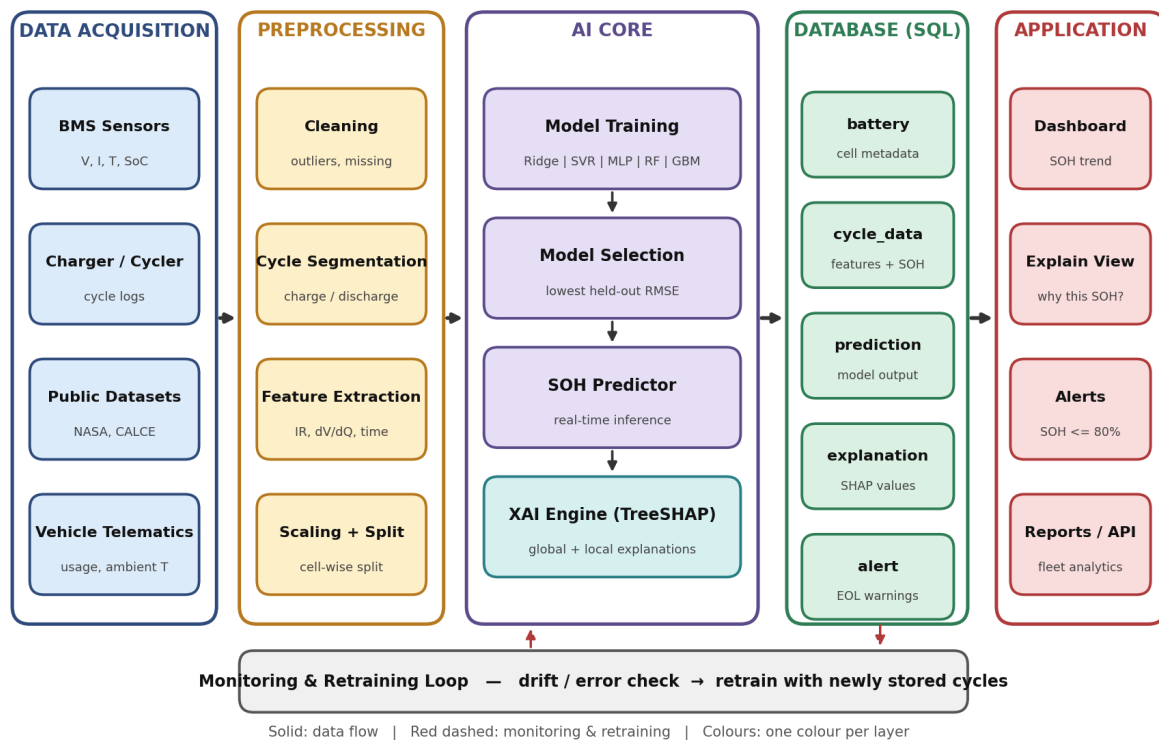


Fig. 4. Proposed system architecture: five layers from data acquisition to application, with a monitoring and retraining loop.

Each transition has a defined contract. For example, preprocessing outputs a feature vector with cell identifier and cycle index; the predictor returns an SOH value and model name; the explainer returns a signed contribution for every feature; and the database layer writes all three with a timestamp. Fig. 5 contrasts the proposed approach with a conventional BMS estimate, and Fig. 6 gives the research workflow.

Conventional BMS Estimation vs. Proposed AI + XAI + Database Framework

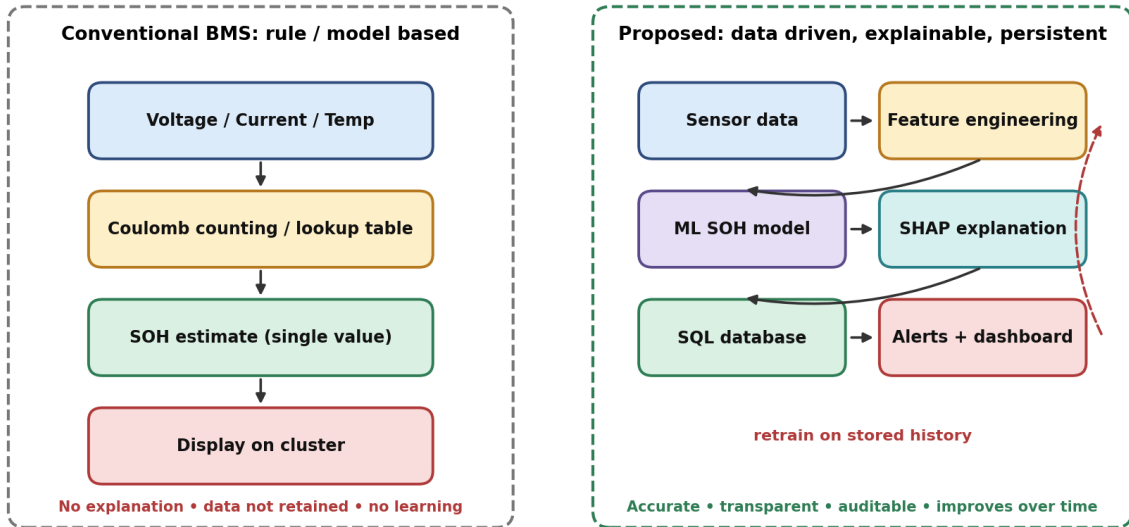


Fig. 5. Conventional BMS estimation compared with the proposed data-driven, explainable and persistent framework.

Research Methodology Flowchart

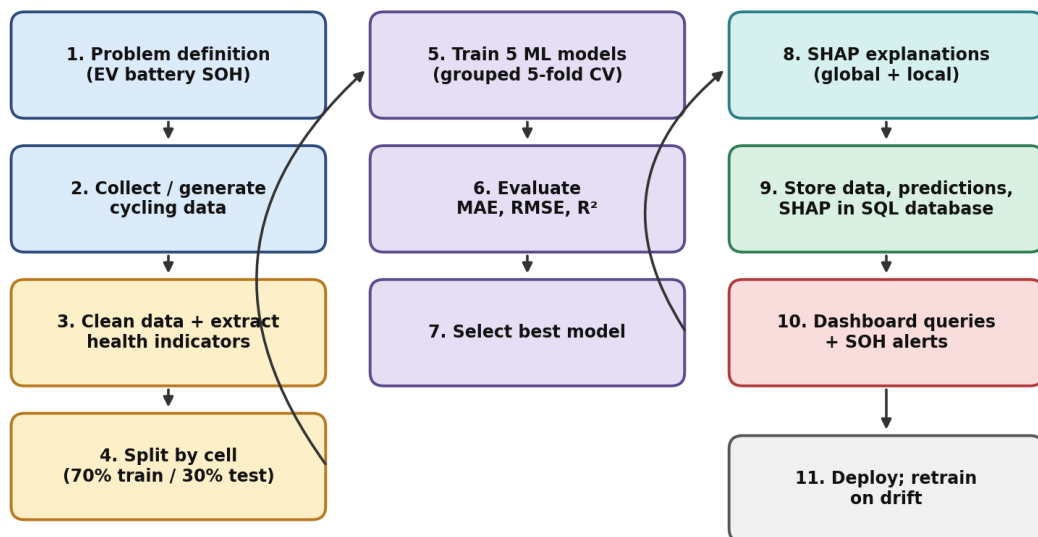


Fig. 6. Research methodology flowchart from problem definition to deployment and retraining.

VI. ALGORITHM AND CONTROL LOGIC

The following procedure describes the training and monitoring workflow. It is model-agnostic and can be implemented as a script, a scheduled job or a service.

TABLE 1. High-level procedure of the proposed framework.

Step	Description
Input	Cycle records D with cell identifier, health indicators X and measured SOH y; test fraction; alert threshold T = 80%.
Preprocess	Clean D; extract health indicators; keep operating conditions as features.
Split	Split D by cell identifier into training cells and unseen test cells (no cell appears in both).
Train	Fit each candidate model on the training cells; run grouped 5-fold cross-validation.
Evaluate	Compute MAE, RMSE and R^2 on the test cells; select the model with the lowest RMSE.
Explain	Run TreeSHAP on the selected model; compute global importance and local contributions.
Store	Insert cell metadata, cycle features, predictions and SHAP values into the SQL database.
Alert	For each cell, raise an alert at the first cycle whose predicted SOH is at or below T.
Monitor	Compare stored predictions with new capacity checks; retrain if error or drift exceeds a limit.

A core invariant is that every stored prediction can be traced to its input features and its explanation. If the model has not seen conditions similar to the input (for example, a different chemistry), the output should be treated as low-confidence and verified by a capacity test.

VII. EXPERIMENTAL SETUP

A. Dataset (Synthetic, Physics-Inspired)

Important disclosure. No real battery measurements were used in this study. To prototype and test the full pipeline, a synthetic cycling dataset was generated by a script that simulates capacity fade and the resulting changes in measurable signals. It contains **40 cells** and **28,201 cycle records**. Each cell has its own ambient temperature (15 to 45 °C), C-rate (0.5, 1.0, 1.5 or 2.0), depth of discharge (70 to 100%), initial capacity (about 2.0 Ah, an 18650-class cell) and initial resistance. Capacity fade per cycle grows with temperature, C-rate and depth of discharge, accelerates after a randomly placed knee, and includes random variation. Internal resistance rises, charge time shortens and the dV/dQ peak height falls as SOH drops, with Gaussian measurement noise added to every signal. Cells are cycled until SOH falls below 62% or a cycle limit of 600 to 900 is reached. The dataset and generator are saved so the experiment can be reproduced.

Because the signals are generated from SOH by simple formulas, the relationships are cleaner than in real cells. Absolute accuracy figures in this paper must therefore not be read as expected real-world performance.

B. Features, Split and Metrics

TABLE 2. Input features used for SOH prediction (target: SOH as a fraction of initial capacity).

Group	Features (12)
Operating conditions	C-rate, depth of discharge (dod), ambient temperature (amb_temp)
Charge / discharge	discharge time (disch_time_min), constant-current charge time (chg_time_min), average voltage (avg_voltage)
Resistance / thermal	internal resistance (internal_res), maximum temperature (max_temp), average temperature (avg_temp)
Ageing indicators	dV/dQ peak height (dvdq_peak), cumulative throughput (ah_throughput), cycle index (cycle)

Data are split by cell: 28 cells (19,584 records) for training and 12 unseen cells (8,617 records) for testing, using a fixed random seed. Grouped 5-fold cross-validation on the training cells provides a robustness check. Metrics are mean absolute error (MAE), root mean squared error (RMSE) and the coefficient of determination R^2 ; errors are reported in SOH fraction and, where noted, in percentage points. Training time and per-sample inference time are also recorded.

Proposed Evaluation Framework

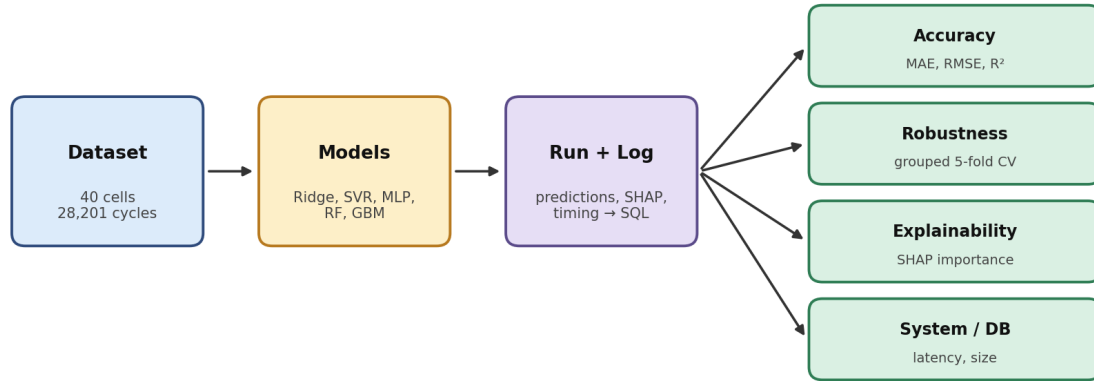


Fig. 7. Evaluation framework: dataset, models, run-and-log stage and four evaluation dimensions.

C. Implementation

The prototype is written in Python with scikit-learn [9] for the models, the SHAP library [1], [2] for explanations and SQLite [12] for the database. All figures and numbers reported in Section VIII were produced by running the code on this dataset in a single session; timings were measured on the machine used and will differ elsewhere.

VIII. RESULTS AND DISCUSSION

A. Data Characteristics

Fig. 8 shows SOH curves of eight sample cells. Cells cycled at higher temperature or C-rate degrade faster, and several show an accelerated decline after the knee. Fig. 9 shows the correlation of each feature with SOH; internal resistance and the cycle index correlate negatively, whereas the dV/dQ peak height correlates positively.

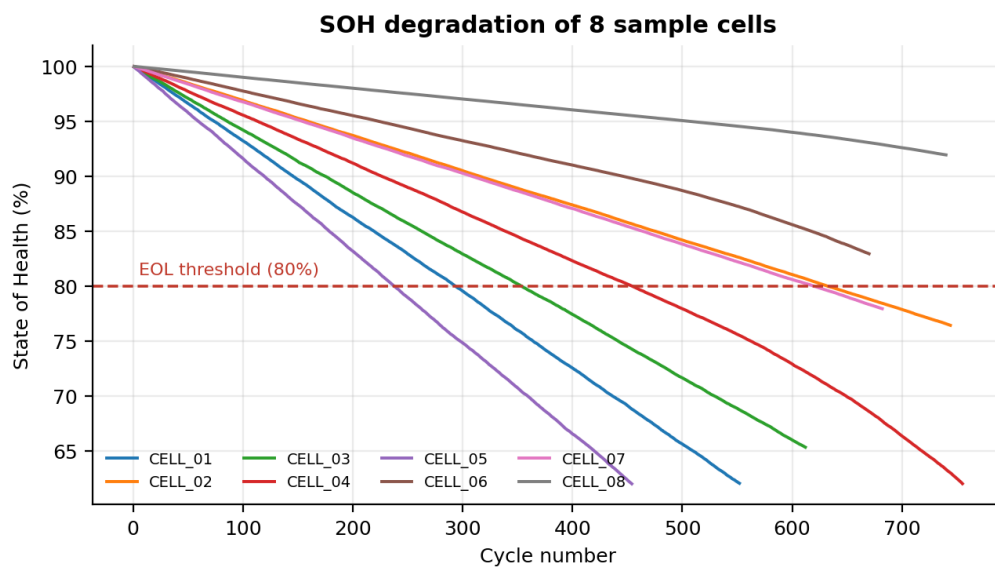


Fig. 8. SOH degradation curves of eight sample cells from the synthetic dataset; the dashed line marks the 80% end-of-life threshold.

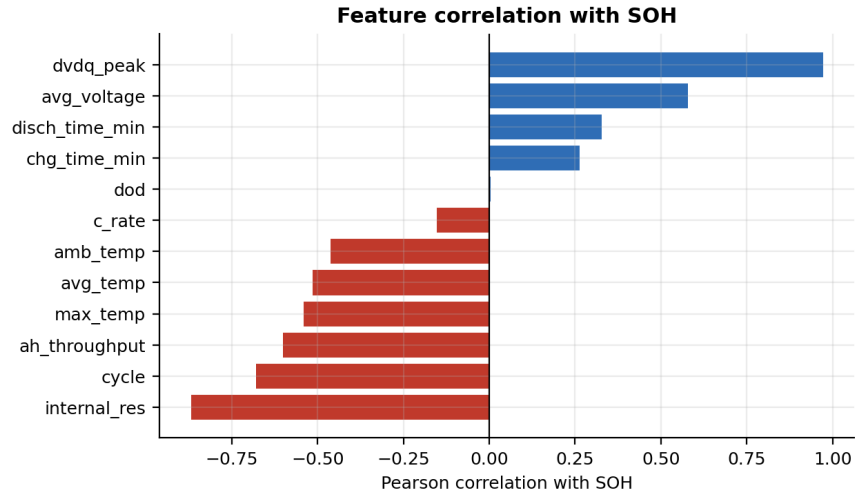


Fig. 9. Pearson correlation of each input feature with SOH.

B. Model Comparison

TABLE 3. Test-set performance on 12 unseen cells (SOH as fraction; 0.01 = 1 percentage point).

Model	MAE	RMSE	R^2	Train time (s)	Inference (ms/sample)
Ridge Regression	0.0140	0.0174	0.9655	0.01	0.00022
SVR (RBF)	0.0211	0.0284	0.9077	1.77	0.01396
MLP (64-32)	0.0418	0.0611	0.5728	4.49	0.00070
Random Forest	0.0156	0.0213	0.9483	34.63	0.02492
Gradient Boosting	0.0126	0.0170	0.9670	15.42	0.00183

Gradient boosting achieved the lowest MAE (0.0126) and RMSE (0.0170) and the highest R^2 (0.967), i.e. an average error of about 1.3 SOH points. Ridge regression was very close (RMSE 0.0174), which is expected because the synthetic signals relate almost linearly to SOH; with such a small gap the ranking of these two models should not be over-interpreted. Random forest followed, SVR was weaker, and the untuned multilayer perceptron performed worst ($R^2 = 0.57$); its result reflects limited tuning and early stopping rather than a general limit of neural networks.

Test-set performance on 12 unseen cells

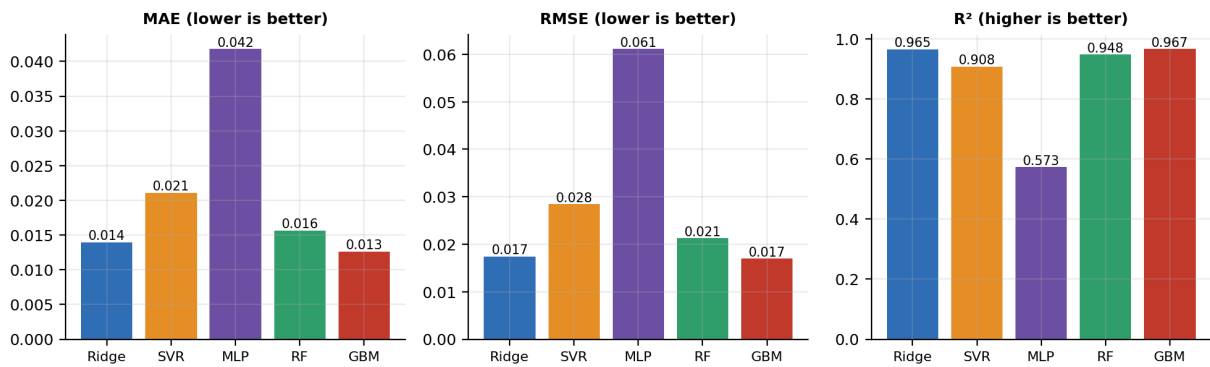


Fig. 10. Comparison of MAE, RMSE and R^2 for the five models on unseen cells.

TABLE 4. Grouped 5-fold cross-validation on the training cells.

Model	CV MAE (mean $\pm$ std)	CV R^2 (mean $\pm$ std)
Ridge Regression	0.0149 $\pm$ 0.0021	0.947 $\pm$ 0.011
SVR (RBF)	0.0273 $\pm$ 0.0086	0.749 $\pm$ 0.210
MLP (64-32)	0.0796 $\pm$ 0.0213	-0.814 $\pm$ 1.252
Random Forest	0.0132 $\pm$ 0.0026	0.945 $\pm$ 0.030
Gradient Boosting	0.0135 $\pm$ 0.0066	0.948 $\pm$ 0.036

Cross-validation tells a slightly different story: random forest (0.0132) and gradient boosting (0.0135) have similar CV MAE, Ridge is a little higher, and the MLP is unstable across folds (negative mean R^2). The fold-to-fold standard deviations show that results depend on which cells are held out, which is why a larger cell population and repeated splits are needed in future work.

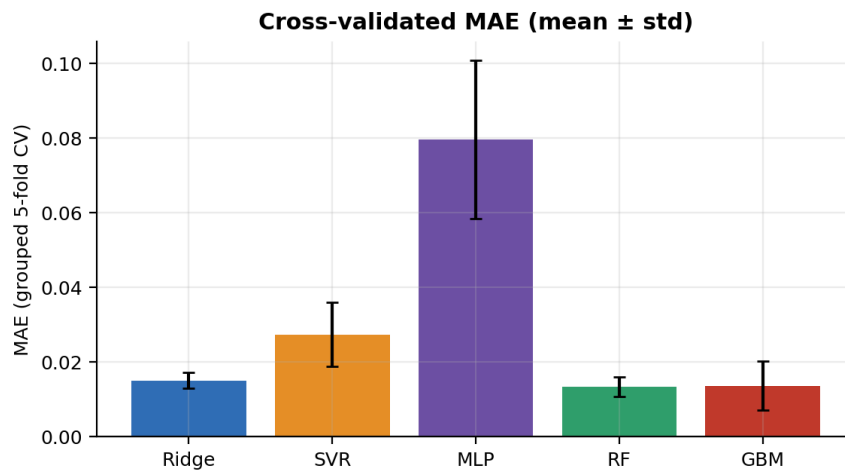


Fig. 11. Cross-validated MAE (mean with standard deviation) for each model.

C. Prediction Quality

Fig. 12 plots predicted against actual SOH for the selected model. Residuals have a mean of +0.47 points and a standard deviation of 1.63 points, so the model slightly overestimates SOH on average; the bias is about +1.1 to +1.2 points for cells between 60 and 80% SOH and close to zero (-0.17) for healthy cells above 90%. Fig. 13 follows one unseen cell (CELL_37) through its life. The prediction follows the downward trend but is noisy and runs above the actual curve in late life, so its first crossing of 80% occurs at cycle 590 against a true crossing at cycle 550 (40 cycles late). Across the test cells, 9 of 12 reached 80% SOH and all 9 triggered an alert, but the alert cycle differed from the true crossing by between -64 and +40 cycles (mean -24, negative meaning early). Early alerts arise mainly because a single noisy prediction can fall below the threshold; smoothing or requiring several consecutive cycles below 80% would be a sensible refinement. Mean absolute error was largest in the 70 to 80% SOH range (1.90 points MAE) and 60 to 70% range (1.82 points), and smallest for healthy cells at 90 to 100% (0.82 points), consistent with the greater variability of late-life behaviour.

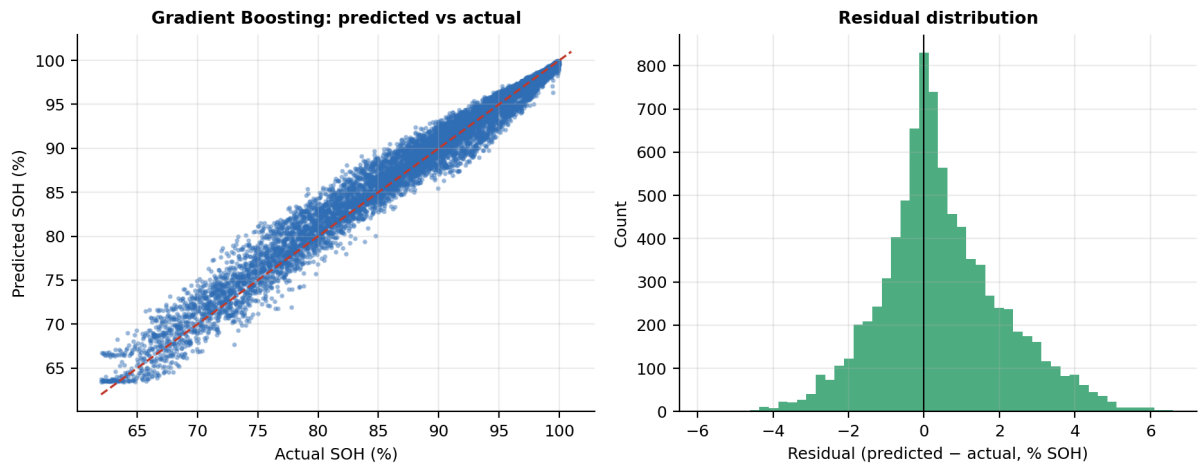


Fig. 12. Selected model (gradient boosting): predicted versus actual SOH on unseen cells, and residual distribution.

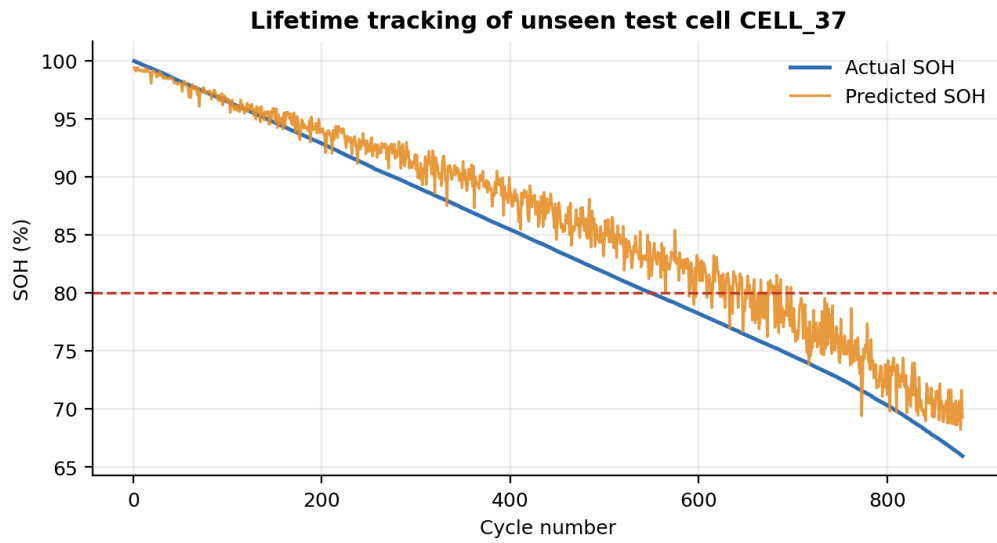


Fig. 13. Predicted versus actual SOH over the life of one unseen test cell, with the 80% alert threshold.

D. Explainability Results

Global SHAP importance (Fig. 14) ranks **dvdq_peak** first (mean $|\text{SHAP}| = 0.043$), followed by **internal_res** (0.015) and **cycle** (0.013); the remaining features each contribute less than 0.004. The beeswarm plot (Fig. 15) shows direction: a high dV/dQ peak pushes the predicted SOH up, whereas high internal resistance and a high cycle count push it down, which agrees with the known physics of ageing. Dependence plots (Fig. 16) show that the effect of the dV/dQ peak is strongest at low values, where the model's SOH estimate drops sharply.

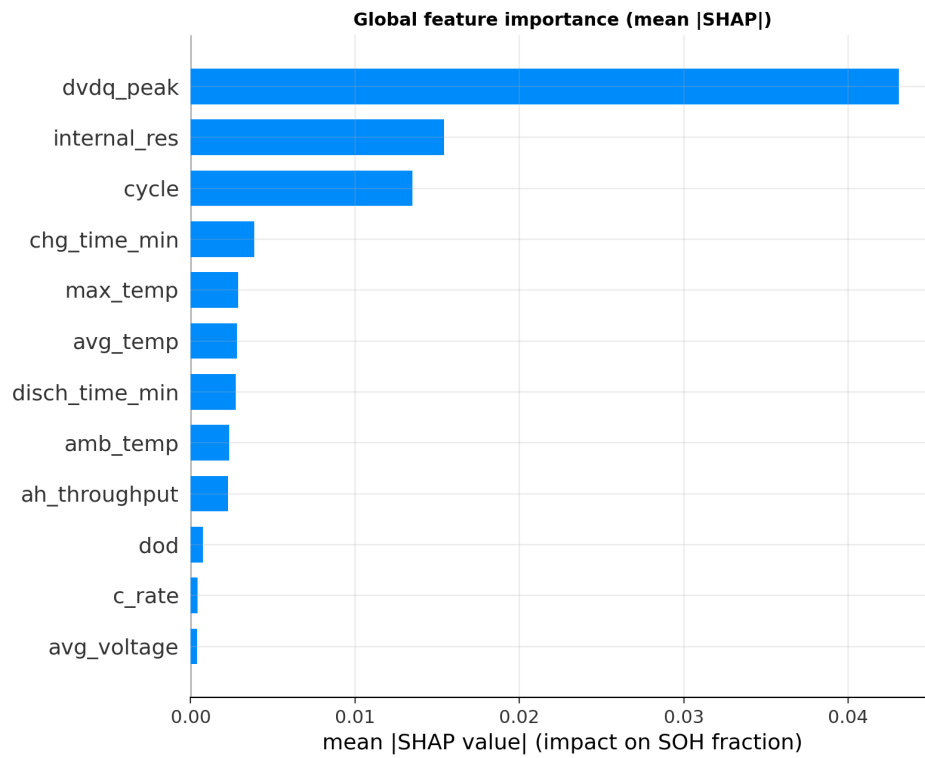


Fig. 14. Global feature importance: mean absolute SHAP value over 600 sampled test records.

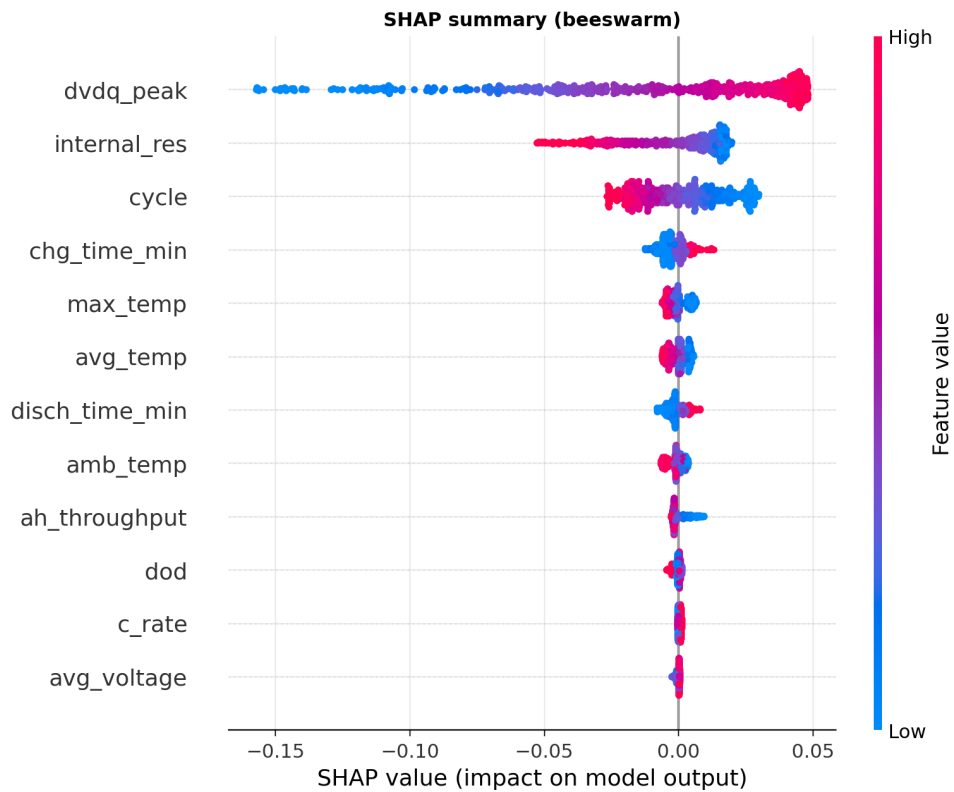


Fig. 15. SHAP beeswarm summary plot. Each dot is one prediction; colour shows the feature value and position shows its effect on predicted SOH.

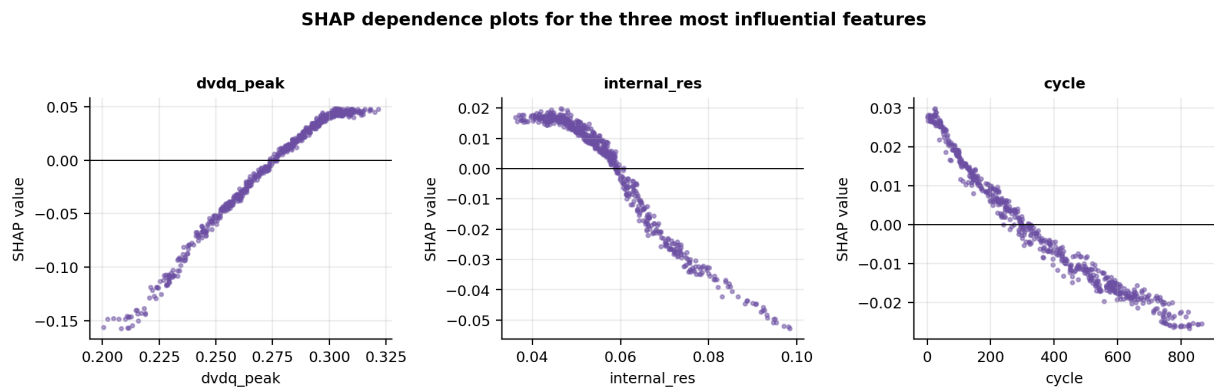


Fig. 16. SHAP dependence plots for dvdq_peak, internal_res and cycle.

A local explanation is shown in Fig. 17 for CELL_18 at cycle 563. The average prediction (baseline) is 89.4% SOH; the model predicts 76.5% (actual 78.0%). The largest contributions are dvdq_peak (-8.1 points), internal_res (-2.7 points), cycle (-1.4 points). Such a breakdown lets an engineer see that the estimate is driven by ageing indicators and not by, for example, ambient temperature alone.

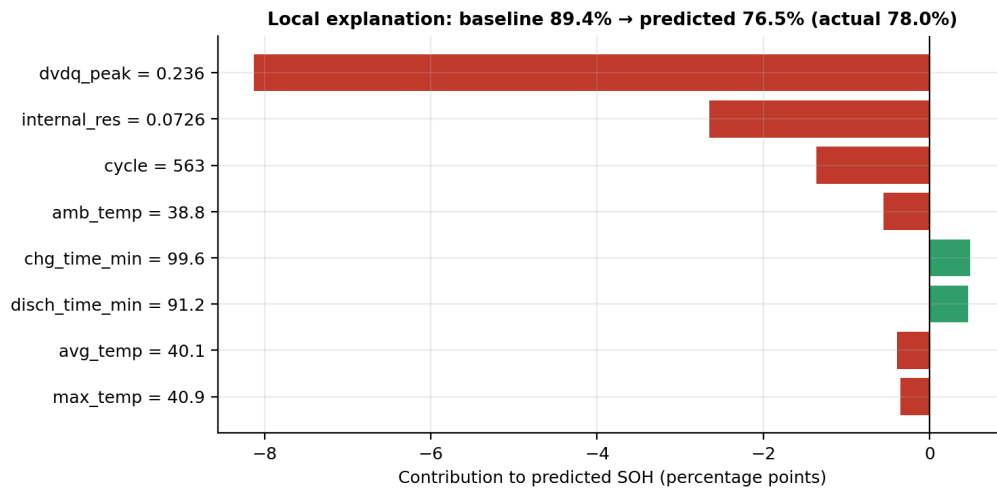


Fig. 17. Local SHAP explanation of a single prediction, showing the eight largest feature contributions.

Caution. The strong role of dvdq_peak and internal_res partly reflects how the synthetic data were generated (both signals are built from SOH). On real cells the ranking may differ, and the SHAP analysis should be repeated and checked against domain knowledge.

E. Database Integration Results

TABLE 5. Contents of the SQLite database after the pipeline run (file size 4.3 MB).

Table	Rows stored
battery	40
cycle_data	28,201
prediction	8,617
explanation (SHAP values)	7,200
alert	9

Bulk insertion of 28,201 cycle records took 0.10 s and insertion of 8,617 predictions took 0.02 s. SHAP values for 600 predictions (12 features each) were stored in the explanation table. The alert rule produced 9 alerts, one for each of 9 of the 12 test cells whose predicted SOH fell to 80% or below. Four representative queries were timed (mean of 200 repetitions):

TABLE 6. Query latency on the prototype database (single machine, SQLite).

ID	Query	Mean latency (ms)
Q1	SOH history of one cell, ordered by cycle	0.217
Q2	Latest prediction per cell (join of prediction and cycle_data)	6.205
Q3	Top-3 SHAP features for one prediction	0.307
Q4	Active critical alerts	0.022

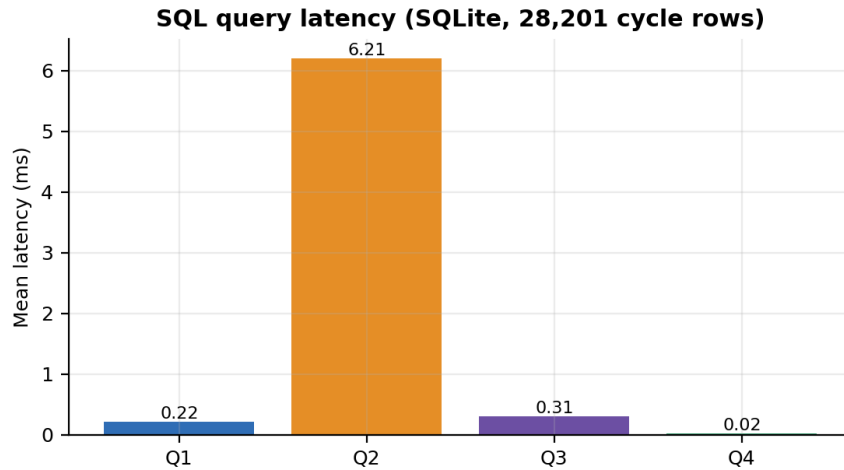


Fig. 18. Mean latency of the four tested SQL queries (Q1 to Q4).

All queries completed in a few milliseconds, with the aggregate join (Q2) being the slowest. This indicates that the schema and indexes are adequate at this scale; behaviour at fleet scale (millions of rows, concurrent users) was not tested and would call for a server database and further tuning.

IX. DESIGN-LEVEL COMPARISON

The table below summarises qualitative design differences between a conventional BMS estimate and the proposed framework. It is a design discussion, not a measured comparison.

TABLE 7. Qualitative design comparison.

Dimension	Conventional BMS estimation	Proposed AI + XAI + database framework
Estimation method	Rules, look-up tables, coulomb counting	Learned from cycle-level health indicators
Adaptivity	Fixed calibration	Retrainable on stored history
Explanation	None	Global and local SHAP explanations
Data retention	Limited or none	Features, predictions, explanations and alerts stored
Auditability	Hard to trace	Each prediction linked to inputs and explanation
Cost / complexity	Low	Higher (training, storage, maintenance)
Evidence in this paper	-	Synthetic data only; real-data validation pending

X. SECURITY, PRIVACY AND GOVERNANCE

Vehicle and battery telemetry can reveal usage patterns and location context, so personal identifiers should be minimised or pseudonymised before storage, and access to the database should be role-based. Model versions and training data snapshots should be recorded so that any alert can be reproduced. SOH predictions should support, not replace, physical capacity tests when decisions affect safety, warranty or resale value.

XI. APPLICATIONS

- Fleet management: ranking vehicles by remaining battery life and scheduling maintenance.
- Warranty and resale: auditable SOH records with explanations for buyers, insurers and service centres.
- Second-life screening: identifying packs suitable for stationary storage.
- Charging and usage advice: showing drivers which habits (temperature, fast charging) accelerate ageing.
- Battery R&D;: using SHAP to compare ageing drivers across chemistries and test conditions.

XII. LIMITATIONS

- **Synthetic data.** All results come from generated data. They show that the pipeline works but do not establish accuracy on real EV batteries.
- **Simplified generator.** Real cells show more complex ageing (lithium plating, knee-point variability, chemistry differences) and noisier signals.
- **Small model tuning effort.** Hyperparameters were set once; the MLP in particular was not tuned.
- **Selection and testing on the same cells.** The best model was chosen from test-set RMSE, which can make its score slightly optimistic; a separate validation split should be used in future work.
- **SHAP reflects the model and the data.** Importances partly mirror how the synthetic signals were built, and correlated features can share credit.
- **Simple alert rule.** The alert fires at the first single prediction at or below 80%, with no smoothing, so it is sensitive to noise (alerts were on average 24 cycles early here, and one cell was 40 cycles late).
- **Prototype database.** SQLite on one machine was used; scalability, concurrency and security were not evaluated.
- **No online monitoring.** The retraining loop is part of the design but was not tested.

XIII. CONCLUSION AND FUTURE SCOPE

This paper presented an AI-based framework for EV battery health prediction that couples machine-learning SOH estimation with SHAP explanations and a relational database. In a cell-wise evaluation on a synthetic dataset, gradient boosting reached an MAE of about 1.3 SOH points with $R^2 = 0.967$; SHAP highlighted the dV/dQ peak, internal resistance and cycle count as the main drivers; and the database layer stored predictions and explanations and answered typical queries within milliseconds. The main value of the work is the integrated, auditable design; the numerical results are a proof of concept.

Future work should (1) validate the pipeline on real datasets such as NASA [4] and CALCE [16] and on pack-level vehicle data; (2) add remaining useful life prediction and uncertainty estimates; (3) compare SHAP with other explanation methods such as LIME [6]; (4) use a separate validation set, systematic hyperparameter search and a smoothed multi-cycle alert rule; (5) deploy the database on a server engine and test fleet-scale load; and (6) implement and evaluate the drift-triggered retraining loop.

REFERENCES

- [1] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [2] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.
- [3] K. A. Severson et al., "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, pp. 383–391, 2019.
- [4] B. Saha and K. Goebel, "Battery Data Set," NASA Ames Prognostics Data Repository, NASA Ames Research Center, 2007.
- [5] Y. Zhang et al., "Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning," *Nature Communications*, vol. 11, art. 1706, 2020.

- [6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," *Proceedings of ACM SIGKDD*, pp. 1135–1144, 2016.
- [7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [10] M. Bercibar et al., "Critical review of state of health estimation methods of Li-ion batteries for real applications," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 572–587, 2016.
- [11] D. Roman, S. Saxena, V. Robu, M. Pecht, and D. Flynn, "Machine learning pipeline for battery state-of-health estimation," *Nature Machine Intelligence*, vol. 3, pp. 447–456, 2021.
- [12] D. R. Hipp et al., "SQLite," software documentation, <https://www.sqlite.org>.
- [13] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [14] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games*, vol. II, Princeton University Press, pp. 307–317, 1953.
- [15] X. Hu, L. Xu, X. Lin, and M. Pecht, "Battery Lifetime Prognostics," *Joule*, vol. 4, no. 2, pp. 310–346, 2020.
- [16] CALCE Battery Research Group, "Battery data," Center for Advanced Life Cycle Engineering, University of Maryland.