vehicles

MDPI

Review

The Double-Edged Sword of Artificial Intelligence in Battery Management Systems for Vehicles: A Critical Review of Progress, Perils, and the Path to Practical Implementation

Claudia Pop †, **Norbert Kertész** † **and Loránd Szabó** *

Department of Electrical Machines and Drives, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; claudia.pop@emd.utcluj.ro (C.P.); norbert.kertesz@campus.utcluj.ro (N.K.)
* Correspondence: lorand.szabo@emd.utcluj.ro
† These authors contributed equally to this work.

Abstract

Artificial intelligence (AI) is increasingly investigated for battery management systems (BMSs) in electric vehicles, but laboratory accuracy does not establish suitability for vehicle use. This critical review examines state estimation, degradation and remaining useful life prediction, fault diagnosis, and thermal management through the gap between reported performance and deployment. A structured narrative synthesis retains the target, dataset, validation split, and execution context of representative results rather than pooling heterogeneous metrics. The analysis addresses data leakage, distribution shift, uncertain labels, calibration, memory, timing, energy use, explainability, safety, cybersecurity, and model updates. Compact learned models, hybrid observers, physical constraints, and runtime monitoring are discussed as responses to these constraints. An author-proposed evidence classification and deployment ladder distinguishes demonstrated validation and implementation from the additional evidence needed for a vehicle decision; neither framework has been independently validated. Industrial sources provide deployment context but do not validate proprietary algorithms. The review finds that practical value requires a demonstrated improvement over a feasible baseline, transfer within a declared domain, measured implementation costs, and verified limits on decision authority. Research priorities include reproducible benchmarks, independent validation, embedded testing, and monitored operation.

Keywords: artificial intelligence; battery management systems; electric vehicles; state of charge; state of health; remaining useful life; TinyML; physics-informed neural networks; explainable AI; functional safety; embedded systems

Check for updates

Academic Editors: Zeyu Chen and Nan Zhou

Received: 5 October 2026
Revised: 22 October 2026
Accepted: 30 October 2026
Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Battery management systems (BMSs) are critical subsystems of modern electric vehicles (EVs), linking electrochemical energy storage to vehicle safety, availability, efficiency, and service life. Their functions include estimating state of charge (SOC; available charge relative to a declared capacity reference), state of health (SOH; battery condition expressed through measures such as capacity or resistance), state of energy (SOE; energy deliverable within specified operating limits), and available charge and discharge power. Remaining useful life (RUL) predicts the time, cycles, or energy throughput before a defined end-of-life (EoL) criterion is reached. These estimates, together with fault diagnosis and thermal management, inform range prediction, torque availability, fast charging, warranty decisions,

second-life assessment, and EoL planning [1,2]. Artificial intelligence (AI), including machine learning (ML) and deep learning (DL), has been proposed to improve estimation and detect degradation or faults earlier. Recent reviews survey these applications across battery systems and BMS functions [3,4]. Whether the reported benefits survive realistic data, vehicle, and embedded-system constraints remains an open question.

The practical problem is that a result obtained from a recorded laboratory trajectory does not establish how an estimator will behave inside a vehicle. Robustness across cells and operating conditions, execution within controller memory and timing budgets, and responses to uncertainty or sensor faults require distinct evidence. Recent reviews discuss these challenges, and embedded experiments demonstrate that model size, arithmetic precision, and target hardware materially affect implementation [5–7]. Safety-related use additionally requires an argument linking the learned function to system requirements, verification, monitoring, and fallback [8].

Table 1 compares the present review with four directly relevant reviews from the literature, highlighting the main similarities and differences in scope, methodology, and contributions. Nazim et al. synthesize AI-based SOH and RUL estimation, including datasets, complexity, and deployment challenges [5]. Shah et al. compare SOC, RUL, and knee-point methods [9]. Nozarijouybari and Fathy examine ML across the battery lifecycle and explicitly discuss tractability, interpretability, robustness, and transferability [3]. Kurucan et al. classify artificial neural network (ANN)-based BMS methods and compare datasets, performance, implementation, and complexity [4]. These reviews already recognize practical limitations; the contribution claimed here is their integration into a common assessment of evidence, execution context, decision authority, and assurance across vehicle BMS functions.

Table 1. Comparison with directly relevant existing reviews.

Review	Coverage	Method	Deployment Perspective	Assurance Emphasis	Contribution Relative to This Review
Nazim et al. [5]	SOH and RUL	Systematic review; AI models and public datasets	Complexity and EV implementation challenges	Prediction reliability and limitations	Task-focused synthesis; extended here to execution evidence and decision authority across functions
Shah et al. [9]	SOC, RUL, knee point	Comparative review of adaptive, learned, and hybrid methods	EV application and future challenges	Accuracy and prognostic limitations	Estimation comparison; complemented here by explicit runtime and fallback requirements
Nozarijouybari and Fathy [3]	Battery lifecycle, from design to management	Broad critical survey; datasets and methodological trade-offs	Tractability, transferability, and robustness	Interpretability and physical/data model integration	Broad battery-ML perspective; narrowed here to vehicle decisions and assurance evidence
Kurucan et al. [4]	ANN-based SOC, SOH, RUL, and diagnosis	ANN taxonomy; datasets, metrics, and reported results	Implementation, complexity, and data availability	Overfitting, accuracy, and model limitations	Method/application classification; extended here by evidence grading and promotion criteria
Present review	Estimation, prognosis, diagnosis, and thermal decisions in vehicle BMSs	Structured critical synthesis; representative quantitative extraction	Measured execution, sensing, data transfer, and updates	Independent monitors, uncertainty, fallback, and assurance case	Links reported benefits and limitations to decision authority using a reusable proposed checklist

This critical review aims to assess when reported AI benefits are supported by evidence sufficient for a defined vehicle BMS use. Its objectives are to identify documented advantages over implementable conventional methods, assess transfer under realistic data and operating conditions, examine embedded resource requirements, evaluate uncertainty and safety controls, and identify the evidence still needed for industrial use. The scope includes

SOC, SOH, degradation and RUL, fault diagnosis, and thermal management; charging and balancing are considered where they define the decision or implementation boundary.

The review contributes a comparison of validation practices across BMS functions, a study-level synthesis that keeps numerical results attached to their datasets and test conditions, and a proposed evidence checklist linked to a staged deployment ladder. The checklist describes the evidence reported by a study; the ladder describes additional evidence proposed before a function acquires operational authority. Both are author-proposed engineering aids, and neither has been externally validated as a certification or prediction instrument. Sections 2–4 define the method and application evidence, Sections 5–7 examine performance, implementation, and assurance, and Sections 8 and 9 develop the architecture and research priorities.

2. Review Scope and Method

This structured critical review uses explicit selection, extraction, and appraisal criteria to examine the relation between algorithmic performance and vehicle use. Relevant reviews differ in target and synthesis method [3–5,9], as compared in Table 1. The heterogeneous tasks, datasets, and evaluation protocols do not support a pooled estimate of algorithm superiority.

2.1. Review Objectives and Questions

The aim and objectives stated in Section 1 were operationalized through five review questions (RQs). They guide the extraction and discussion below and are answered concisely in Section 10.

RQ1. *For which BMS functions does AI provide a documented advantage over conventional estimation, diagnostic, or control methods?*

RQ2. *How robust are the reported results across cells, chemistries, temperatures, duty cycles, ageing paths, and measurement conditions?*

RQ3. *What computational, memory, latency, energy, and implementation evidence is reported for onboard or edge deployment?*

RQ4. *How do studies address uncertainty, interpretability, physical consistency, fault behavior, cybersecurity, and safe fallback operation?*

RQ5. *Which technical and validation gaps currently prevent a transition from laboratory demonstrations to production vehicle BMSs?*

2.2. Search Scope and Source Identification

The core publication window was January 2019–September 2026, covering recent growth in data-driven BMS research. Earlier sources were retained selectively to establish definitions and conventional estimation practice [10–12], foundational Kalman filtering [13], equivalent-circuit models [14], and ageing mechanisms [15]. Established appraisal instruments were also retained for methodological provenance. These sources provide the physical and methodological baseline for interpreting later AI studies; they do not represent current algorithmic performance or deployment readiness.

Literature was identified through Web of Science, Scopus, and the Institute of Electrical and Electronics Engineers (IEEE) Xplore database, supplemented by citation tracing and targeted retrieval of standards, public datasets, and primary industrial sources.

Search expressions combined four keyword families. The system family included BMS, EV, lithium-ion battery, battery pack, and battery module. The AI family included AI, ML, DL, neural network, transformer, Gaussian process, explainable AI, physics-informed neural network, digital twin (DT), and tiny machine learning (TinyML). The function family

included SOC, SOH, SOE, RUL, knee point, fault diagnosis, thermal management, charging, and cell balancing. The deployment family included embedded, microcontroller, onboard, real-time, hardware acceleration, quantization, pruning, uncertainty, robustness, safety, validation, and cybersecurity. Families were connected with AND, and synonyms within a family with OR; combinations were adapted to the topic and source. The scope extends beyond the SOH/RUL focus of Nazim et al. [5] to implementation, diagnosis, thermal decisions, and assurance.

A prospectively registered systematic-review protocol and a complete screening log are not available for this critical review. Exhaustive coverage and screening-stage record counts are therefore not claimed, and missing counts have not been reconstructed. The safeguards are explicit eligibility rules, traceable source attribution, disclosure of validation context, and separation of reported results from engineering interpretation. These choices limit reproducibility of source selection and may introduce selection bias; the evidence tables support critical comparison rather than prevalence estimates or meta-analysis.

2.3. Eligibility Criteria

A source was included when it satisfied at least one of the following criteria:

- It estimated or predicted a BMS-relevant state or health indicator, including SOC, SOH, SOE, power capability, RUL, or degradation knee point.
- It diagnosed a safety-relevant battery or sensor fault, predicted an abnormal condition, or supported early warning and protection decisions.
- It optimized or controlled a BMS-relevant process such as charging, thermal management, cell balancing, or energy and power allocation.
- It demonstrated an implementation constraint or enabling method directly relevant to vehicle deployment, including TinyML, model compression, hardware acceleration, uncertainty quantification, explainability, physics-based constraints, or runtime monitoring.
- It provided a review, methodological appraisal instrument, automotive standard, regulation, or primary industrial source needed to interpret validation, deployment, or assurance. These contextual sources were distinguished from experiments on battery algorithms; no battery-performance evidence class was assigned to them.

Peer-reviewed journal articles were prioritized; conference papers were retained when they contributed distinct implementation evidence. Standards and primary industrial documents were considered separately from algorithmic experiments. Non-vehicle studies were retained when their measurements or resource constraints had a clear transfer rationale. For example, unmanned aerial vehicle (UAV) TinyML prognosis provides embedded evidence in a different application [16], whereas smartphone short-circuit detection provides experiments on held-out batteries with inference evaluated on a computer [17]. Neither establishes vehicle BMS validation.

Sources were excluded when any of the following conditions applied:

- The work addressed materials discovery, cell design, or charging infrastructure without a clear connection to a BMS state, decision, safety function, or implementation constraint.
- The article provided only a conceptual architecture, promotional claim, or algorithm description without sufficient information about data, validation, or operating conditions.
- The study duplicated a later peer-reviewed version, lacked retrievable bibliographic information, or did not provide enough methodological detail for evidence grading.
- The reported result concerned a non-rechargeable system or a battery application whose measurements, operating regime, and implementation requirements were not meaningfully transferable to vehicle BMSs.

- The full text was unavailable, and the abstract did not contain enough information to support a specific technical claim; such records could inform source discovery but were not used for quantitative comparison.

2.4. Study Classification and Data Extraction

Each source was classified by publication type and by the level at which the evidence was generated. Publication types comprised experimental studies, simulation-only studies, mixed experimental and simulation studies, reviews, standards, and primary industrial documents. Experimental work was further separated into cell, module, pack, and vehicle levels. Implementation location was coded as offline workstation analysis, cloud or fleet analysis, software-in-the-loop (SIL), processor-in-the-loop (PIL), microcontroller or edge implementation, hardware-in-the-loop (HIL), or in-vehicle demonstration. This distinction is essential because a model evaluated offline using laboratory cycling data is subject to fundamentally different timing, memory, energy consumption, update, and fault-handling constraints than an estimator deployed within a vehicle controller [6,7,18].

For the primary studies examined in detail, extraction considered the BMS task and target; chemistry, format, and scale; data source and experimental units; temperature and current; charging or duty cycle; input window and forecast horizon; model and baseline; split and adaptation strategy; uncertainty; and implementation resources. The paper provides in Section 5 a representative study-level extraction, including source locations and limitations on interpretation. Missing details are identified explicitly rather than inferred from a model label. Physics-informed, hybrid, and uncertainty-aware studies were additionally assessed for the physical relation, ablation, and calibration evidence [19–25].

2.5. Evidence Grading

The review records E1–E3 data and validation classes together with an independent E4 implementation or operational qualifier. E1–E3 describe the evidence supporting a result; E4 describes where execution or operation was demonstrated. These descriptors differ from the proposed L1–L5 deployment stages in Section 8.2 and do not constitute a general study-quality score.

- E1: simulated or synthetic evidence. Performance is evaluated only on simulated signals, a digital model, or synthetic faults, without a corresponding experimental test.
- E2: controlled laboratory evidence. Experimental measurements are used, but validation is limited to within-cell, within-protocol, or another split that does not establish transfer to independent cells or conditions. The validation unit and any shared information between development and testing must be recorded.
- E3: cross-cell or cross-condition evidence. The evaluation excludes identified cells, protocols, temperatures, chemistries, or datasets from model development. The excluded axis and any target-domain adaptation must be stated; one form of holdout does not establish transfer along other axes.
- E4: implementation or operational qualifier. E4-hardware records measured execution on the intended controller or a documented target-equivalent platform; E4-pack and E4-field identify demonstrated module/pack and vehicle/fleet settings, respectively. Record the final preprocessing, runtime, precision, memory, timing, workload, and any unreported items. PIL or HIL provides hardware evidence only when the stated target is actually in the loop. A proposed architecture or timing measured only on a desktop or graphics processing unit (GPU) does not qualify as E4-hardware. The qualifier records a demonstration, while the fuller acceptance requirements for deployment are assessed separately.

The scheme was developed for this review and has not undergone independent validation, an inter-rater reliability study, or prospective testing against deployment outcomes. Its author-assigned descriptors complement methodological appraisal. The Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis plus AI (TRIPOD + AI) statement provides reporting guidance [26], while the Prediction model Risk Of Bias ASsessment Tool plus AI (PROBAST + AI) addresses quality, risk of bias, and applicability [27]. Both concern prediction models in healthcare. Their attention to data, targets, analysis, and validation informs the appraisal questions here, but their clinical criteria are not treated as validated automotive instruments.

Review-level appraisal addresses a different unit of assessment. The updated A Measurement Tool to Assess Systematic Reviews (AMSTAR 2) appraises the methodological quality of systematic reviews [28], whereas Risk Of Bias In Systematic reviews (ROBIS) focuses on bias in review methods and conclusions [29]. AMSTAR 2 extends the original AMSTAR instrument [30]. These healthcare tools are cited as methodological analogues; they do not validate the primary-study evidence classification proposed here.

A study may therefore be recorded as E3 + E4-hardware or E2 + E4-pack. An implementation qualifier cannot compensate for leakage, unreliable labels, weak calibration, or untested transfer; E3 laboratory evidence can establish a specified transfer claim without demonstrating embedded execution. Appendix A provides separate evidence and appraisal checklists, including rules for incomplete or mixed evidence.

2.6. Metric Normalization and Critical Synthesis

The literature was synthesized narratively and through structured comparison tables. Mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), mean absolute percentage error (MAPE), maximum absolute error, classification accuracy, precision, recall, F1 score (the harmonic mean of precision and recall), detection delay, and interval coverage were extracted as reported. Direct numerical comparisons were made only when the target definition, scaling, operating window, prediction horizon, and evaluation split were sufficiently similar. In particular, an SOH error derived from capacity during a complete reference cycle was not treated as equivalent to an error obtained from a short and randomly located charging segment [23,25,31]. Results from random cycle-level splits were distinguished from cell-level holdout tests because temporally or degradation-wise correlated samples from the same cell or degradation trajectory may occur in both training and test sets.

Reported accuracy was interpreted together with uncertainty, generalization, and resource use. A lower average error did not automatically receive a stronger assessment when it was accompanied by weak external validation, absent confidence calibration, an undisclosed model footprint, or no safe response to out-of-distribution inputs. For fault-diagnosis studies, false-negative behavior and fault coverage were given greater weight than aggregate accuracy [17,32]. For explainable models and DT, the review recorded whether explanations were used to test model behavior, detect spurious correlations, or support an engineering assurance argument [33,34]. Safety and regulatory statements were checked against the scope of the cited standards, including the publicly available specification (PAS) ISO/PAS 8800:2024 from the International Organization for Standardization (ISO), which addresses AI elements in road-vehicle safety-related systems [8].

The synthesis keeps three judgments distinct: the reported data/validation class (E1–E3) and any E4 implementation qualifier, methodological limitations, and the deployment gates proposed in Section 8.2. Experimental pack data can support E4-pack while leaving processor timing or fallback untested. This separation prevents a realistic data source or compact model from being interpreted as evidence of production readiness.

3. Battery Management Context and Deployment Boundary

This section defines the measurements, latent states, hierarchy, and operational boundary against which later AI claims are judged.

3.1. Measured Variables and Inferred States

A vehicle BMS integrates measurement, estimation, protection, and control functions, continuously monitoring battery states, estimating key parameters, enforcing safety limits, and supporting operational decision-making [35]. These measurements are affected by sensor offset, quantization, sampling delay, electromagnetic interference, wiring-induced voltage drops, thermal gradients, and occasional missing or implausible values [1,10]. Hardware architecture therefore matters: the number and placement of sensors, the partition between monitoring units and the BMS master, the communication bus, isolation, redundancy, contactor control, and balancing circuitry determine what information is available to any estimation algorithm [36,37].

Most quantities needed for vehicle control cannot be measured directly during normal operation. SOC, SOE, SOH, state of available power, internal temperature, degradation mode, and RUL are latent or model-dependent quantities inferred from measured signals and prior information. The inference chain usually contains a battery model, parameter identification, and a state observer or data-driven estimator. Equivalent-circuit models remain important because they offer a workable compromise between fidelity and computational cost [2,14]. Kalman-filter and BMS reviews show how measurements and a dynamic model are reconciled recursively [1,13]. These conventional estimators provide the important physical and computational reference against which an AI method should be judged.

This baseline role does not mean that conventional observers are failure-free. Equivalent-circuit models and Kalman-filter variants can suffer from parameter drift, model-order mismatch, poor observability in flat open-circuit-voltage regions, sensor bias, and incorrect noise assumptions. The critical comparison in this review therefore treats conventional methods as established and auditable reference points, not as intrinsically safe defaults; AI methods should be judged by whether they improve a declared function without weakening boundedness, diagnosability, or fallback behavior.

The same voltage–current–temperature history does not have an invariant interpretation across chemistries or operating conditions. Open-circuit-voltage slopes, hysteresis, diffusion dynamics, resistance, heat generation, and ageing mechanisms vary with chemistry, cell format, temperature, SOC region, current history, and prior degradation [10,38]. A flat open-circuit-voltage region, for example, reduces the information available for voltage-based SOC correction, while temperature and ageing alter both usable capacity and dynamic response. An estimator trained on one chemistry or protocol can therefore be precise within its calibration domain and biased outside it [12,15,39].

3.2. Operational States and Their Decision Meaning

SOC is commonly expressed as the remaining charge relative to a reference capacity, but that reference is not constant over battery life and is not equivalent to the energy available for propulsion. Coulomb counting accumulates current-sensor and initialization errors, whereas voltage- or model-based correction depends on rest conditions, model validity, and identifiable voltage response [10,12]. SOC should therefore be reported with its reference capacity, sign convention, initialization method, and operating range. A small average SOC error does not demonstrate that the estimator recovers from bias or remains bounded during long dynamic operation [13,39].

SOE expresses the energy that can be delivered or accepted over a defined operating window. It depends on the voltage trajectory as well as charge and is therefore sensitive to

temperature, load, SOC limits, ageing, and auxiliary power demand. State of power (SOP), also described as state of available power, describes the maximum charge or discharge power that can be sustained for a specified horizon without violating voltage, current, temperature, or SOC constraints. It is a time-dependent capability rather than a static cell rating. The forecast horizon and limiting constraint must accompany any reported value because acceleration, regenerative braking, fast charging, and limp-home operation impose different requirements [1,2,40].

SOH is not a single universally defined state. Capacity-based SOH, resistance- or impedance-based SOH, power capability, and application-specific performance loss describe related but non-equivalent aspects of ageing. Capacity measured during a complete reference test is different from capacity inferred onboard from partial and irregular operation [11,23]. Because capacity fade and resistance growth may progress at different rates, one scalar health value can conceal a loss of power capability or a change in safety margin [15,41,42]. For this review, every SOH result is interpreted using the definition of the health target used in the cited study, reference test, temperature, current rate, and EoL criterion.

A degradation trajectory describes the time evolution of capacity, resistance, power capability, or a related health indicator. RUL denotes the future time, cycles, energy throughput, or distance until a specified threshold is reached. It is conditional on the future duty cycle and chosen EoL threshold, not an intrinsic property that can be labelled independently of use [21,22]. A fault state is different again: it represents an abnormal condition that may require immediate detection and protection, such as an internal short circuit, sensor failure, isolation loss, contactor fault, thermal anomaly, or cell inconsistency [17,32]. Prognosis and fault diagnosis should not be merged into a single accuracy ranking because their time scales and error costs differ.

Table 2 therefore defines the main BMS targets used throughout the critical comparison and records the minimum context needed before reported errors can be interpreted.

Table 2. BMS functions and target definitions used for critical comparison.

Function or Target	Quantity or Event Inferred	Vehicle Decision Affected	Minimum Evaluation Context
SOC	Available charge relative to a declared capacity reference	Range, power limits, balancing, charge control	Initialization, causal update, temperature, ageing, bias recovery
SOE	Deliverable or absorbable energy over a declared voltage, SOC, temperature, and load window	Range prediction, charging limits, reserve planning	Voltage trajectory, energy window, auxiliary loads, temperature, ageing, operating limits
SOP	Maximum charge or discharge power over a specified horizon without violating limits	Torque availability, regenerative braking, derating, fast charging	Forecast horizon, limiting constraint, thermal state, resistance, pack imbalance
SOH	Capacity, resistance, power capability, or application-specific health	Warranty, maintenance, charging and power margins	Target definition, reference test, cell/pack holdout, temperature
RUL or trajectory	Future degradation path and threshold-crossing time	Inspection, retirement, second life, reserve planning	Forecast origin, horizon, duty scenario, threshold, interval calibration
Fault diagnosis	Abnormal electrical, thermal, sensor, connection, or isolation condition	Alarm, derating, isolation, shutdown, service	Event holdout, severity, delay, false alarms per operating time
Charging, thermal and balancing control	Action that optimizes a constrained operating objective	Charge time, temperature, ageing, cell consistency	Closed-loop stability, constraints, actuator limits, safety fallback

3.3. Cell, Module, and Pack Hierarchy

Many laboratory studies treat a single cell as the complete system. In practice, vehicle batteries are hierarchical packs of interconnected cells, modules, sensors, electronics, and thermal management components. Because cells differ in capacity, resistance, temperature, self-discharge, and ageing, pack behavior cannot be reliably inferred from an average cell. Accurate pack-level prediction therefore requires explicit consideration of cell variability and imbalance [36,43].

The most constrained cell can determine the usable pack envelope because charging or discharging must stop when any monitored cell reaches its voltage, current, or temperature limit. Balancing can reduce SOC dispersion, but it does not eliminate differences in capacity, resistance, or thermal history. Passive and active balancing also impose different losses, hardware requirements, switching constraints, and diagnostic observability requirements [36,44]. An AI model evaluated on a single cell should therefore not be described as a pack-level BMS solution unless the method has been tested against cell variation, balancing activity, sensor topology, and pack protection logic.

Thermal state is likewise distributed. Surface temperature at a limited set of sensor locations does not reveal every cell's core temperature or local hot spot, and non-uniform cooling can accelerate divergence among cells. Thermal management must keep cells within permissible limits while controlling gradients and auxiliary energy consumption. The coupling between temperature, resistance, power limits, degradation, and fault development means that electrical and thermal estimators cannot be validated independently when their outputs drive the same operating decision [45–47].

3.4. Onboard, Cloud-Assisted, and Integrated Battery Management Systems

An onboard BMS executes functions required to keep the battery and vehicle operating within defined safety and performance limits. It acquires signals, checks plausibility, estimates fast-changing states, controls contactors and balancing, communicates limits to the vehicle controller and charger, and initiates protective action. These functions operate under bounded memory, processor, communication, energy, and timing resources. Safety-relevant loops must continue during loss of connectivity and must have deterministic behavior, diagnostic coverage, and a defined fallback. Hardware topology and the division between slave monitoring units and the BMS master therefore constrain the feasible model before algorithm selection begins [36,37].

A cloud-assisted BMS moves selected storage and computation to a remote service. Fleet data can support long-horizon degradation analysis, individualized model updates, anomaly screening across vehicles, and comparison with similar batteries. DT and cloud-side-end architectures demonstrate how local measurements may be combined with cloud models for SOC and SOH estimation or lifecycle management [34,48,49]. The cloud, however, introduces transmission delay, intermittent connectivity, privacy and cybersecurity obligations, version control, and a dependence on data quality. It is therefore well suited to advisory diagnostics and model maintenance, but it should not be treated as the sole path for an immediate protection decision.

A functionally integrated BMS exchanges information, operational constraints, and control requests with the vehicle energy-management controller, charger, thermal-management system, inverter, service diagnostics, and fleet backend. Integration can improve usable energy and battery lifetime by coordinating current and power limits with route planning, charging strategies, and thermal-management objectives. However, it also enlarges the assurance boundary, as an erroneous estimate may affect torque availability, charging current, cooling demand, warranty decisions, or maintenance scheduling. Interface timing, signal

ownership, fault containment, and the authority to override a learned recommendation should therefore be explicitly specified as part of the AI use case [1,2,37].

3.5. Deployment Boundary for Artificial Intelligence

The deployment boundary determines what an AI result means. Offline models can use complete cycles, future information, extensive preprocessing, and workstation-class computing. Fleet models can aggregate many vehicles and can be retrained after deployment. An onboard estimator must work causally with the sensors, sampling rate, memory, latency, and arithmetic available in the target controller. A model that estimates SOH accurately from a complete reference discharge may be useful for laboratory characterization or service diagnostics but unusable for continuous vehicle operation. Likewise, a cloud anomaly score can prioritize inspection without being suitable for opening a contactor.

The following decision layers guide interpretation of the applications in Section 4: laboratory characterization, offline or fleet analytics, onboard advisory estimation, supervisory control, or safety-relevant protection. The evidence required becomes stricter as the output moves closer to direct control. At minimum, an onboard method should disclose input availability, initialization, update rate, worst-case latency, memory and compute footprint, response to missing or biased sensors, validated operating domain, uncertainty or confidence logic, and the conventional fallback. Safety relevance also requires a clear account of how the learned output is checked against physical and operational limits [3,8].

This boundary prevents two common category errors. First, cell-level offline accuracy is not evidence of pack-level vehicle readiness. Second, access to cloud computing does not remove the need for local protection and fail-safe behavior. AI can provide valuable residual correction, latent-state inference, prognosis, and anomaly detection, but its practical role depends on where the model executes, which decision consumes its output, and what independent mechanism preserves safety when the model is uncertain or unavailable. Figure 1 summarizes the deployment boundary established in this section and makes the independent checks and fallback path explicit.

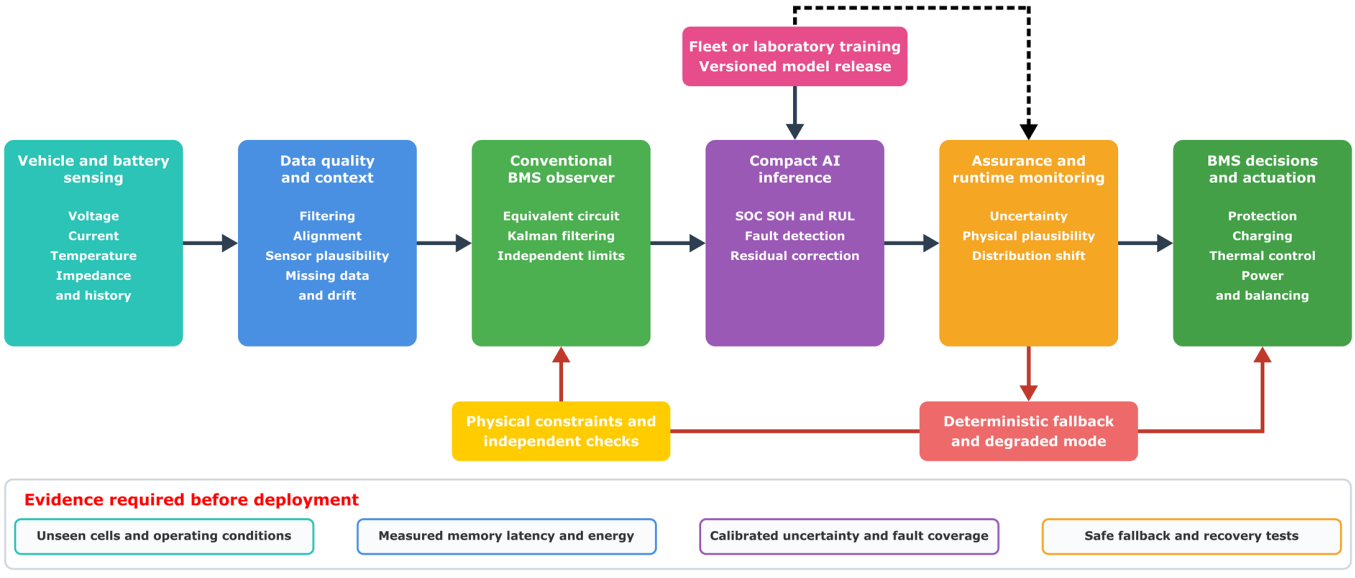


Figure 1. Proposed deployment stack for an AI-enabled vehicle BMS. Learned estimation, prognosis, or fault-detection functions remain bounded by data-quality checks, a conventional observer or physical limits, runtime assurance, and deterministic fallback; cloud or laboratory services support governed model development and release.

4. Artificial Intelligence Applications in Battery Management Systems

The five application domains below are assessed by target definition, data provenance, generalization, uncertainty, implementation evidence, and the decision affected in the vehicle.

4.1. State of Charge Estimation

SOC estimation transforms causal current, voltage, temperature, and recent history into a bounded charge estimate during charging, discharging, rest, and regeneration. Its output affects range, power limits, charging, and balancing [3,9]. Established estimation reviews discuss initialization, convergence, and drift [12,39]. General modeling reviews and Plett's Kalman-filter framework provide conventional reference points [1,2,13]. AI application reviews and measured embedded studies add the data and resource constraints relevant to learned estimators [4,6,7].

Feed-forward neural networks (FFNNs), including deep neural networks (DNNs), learn mappings from finite measurement windows to SOC; recurrent models also represent temporal dependence. Long short-term memory (LSTM) and gated recurrent unit (GRU) are recurrent architectures used for sequence learning. SOC studies include LSTM and DNN estimators [9,50,51], while convolutional neural networks (CNNs) appear in compact embedded implementations [6,7]. Bidirectional models use context on both sides of a sample [52,53]. If future samples are required, the estimator is non-causal unless a corresponding output delay is disclosed; whole-trajectory smoothing does not establish instantaneous onboard estimation.

Reported SOC errors are strongly conditioned by the dataset and split. The widely reused Panasonic and LG cylindrical-cell datasets provide valuable common baselines [50,51,54], but repeated windows from the same cell, temperature, and drive cycle are highly correlated [52,53]. A random sample split can therefore place nearly adjacent observations in training and testing, and a model may implicitly learn the reference Coulomb-counting trajectory used to generate its labels. Stronger evidence holds out complete drive cycles and cells, varies the initial SOC, separates temperatures, and tests ageing states that were not used during training. Chemistry, nominal capacity, sampling interval, input-window length, filtering, label construction, and any use of future measurements must be reported before MAE or RMSE values are compared.

Hybrid estimators combine a learned component with a recursive observer or filter. Examples include LSTM outputs smoothed or corrected by adaptive cubature and unscented Kalman filters [55,56]. This architecture can improve noise rejection and convergence from an incorrect initial SOC, and it offers a natural location for covariance and plausibility checks. The benefit is not automatic, however. If the network target and filter observation are derived from the same current and voltage history, their errors are correlated; treating them as independent can produce overconfident updates. A hybrid method should therefore disclose the process and measurement models, covariance adaptation, reset logic, physical bounds, and behavior when the learned estimate disagrees persistently with Coulomb counting or the voltage model.

Transfer learning and rolling updates aim to reduce the data required for a new cell and to compensate for ageing, while attention-based sequence models aim to retain information from longer windows [57–59]. These methods address real limitations associated with a fixed model under cell-to-cell variation and ageing, but they also create configuration-control questions. The source and target chemistries, amount of target data, frozen and retrained layers, update trigger, validation set, and rollback rule should be stated. Fine-tuning on a small sample from the test battery is adaptation, not zero-shot generalization, and a result on another laboratory cell does not by itself establish pack-level performance under sensor tolerances, cell imbalance, or thermal gradients.

Measured embedded execution provides a more direct implementation test than model size. Mazzi et al. and Pau and Aniballi report execution on microcontroller units (MCUs) [6,7]. Giazitzis et al. report compact projected models, but their latency measurements use an RTX 3060 GPU [18]. Resource evidence should distinguish 32-bit floating-point (FP32) results from 8-bit integer (INT8) deployment results and report flash, peak random-access memory (RAM), buffers, precision, latency, update rate, energy, compiler, and runtime for the tested configuration. Timing jitter, missing samples, saturation, and fallback require testing on the intended target.

For vehicle use, SOC evaluation should emphasize causal updates, recovery from initialization and sensor bias, and bounded drift over long operation. Where the intended decision requires it, the system should detect or flag departure from its validated operating domain. The common implementation and assurance requirements are consolidated in Table 3 below, and further detailed in Tables 5 (Section 5) and 6 (Section 6.4), as well as in Appendix A; an offline average error alone cannot establish readiness.

Table 3. Summary of AI advantages, evidence requirements, and deployment risks across BMS application areas.

Application Area	Potential AI Advantage	Evidence Before Strong Claim	Deployment Risk
SOC estimation	Captures nonlinear voltage–current–temperature history and improves observer initialization, bias recovery, and transient response.	Causal inputs; cell or drive-cycle holdout; ageing and temperature variation; measured target-hardware latency and memory.	Look-ahead leakage; correlated observer errors; timing overruns; drift under sensor or domain shift.
SOH estimation	Fuses partial-charge, impedance, resistance, and fleet signals into an auditable health estimate.	Declared SOH target and label source; cell or pack holdout; cross-condition transfer; calibrated uncertainty.	Ambiguous health definition; label leakage; weak transfer across chemistries or protocols; unavailable onboard measurements.
RUL and degradation trajectory	Forecasts threshold crossing for maintenance, warranty, and retirement decisions.	Declared threshold and forecast origin; separated train and test cells; future-use assumption; horizon calibration; knee and EoL evaluation.	Retrospective reconstruction mistaken for prognosis; protocol dependence; poor knee prediction; uncertainty under unknown future use.
Fault diagnosis and safety-relevant detection	Detects abnormal residuals, rare events, and multi-sensor patterns.	Event-level holdout; severity-stratified delay; false alarms per operating hour; open-set behavior; verified response path.	Class imbalance; synthetic faults; missed hazards hidden by aggregate accuracy; unclear intervention authority.
Thermal management	Estimates hidden thermal states and supports predictive cooling and charging decisions.	Independent thermal tests; ageing and fast-charge variation; temperature/energy trade-off; measured closed-loop execution.	Unobserved hot spots; wrong thermal parameters; actuator saturation; temperature improvement at excessive energy cost.

4.2. State of Health Estimation

SOH estimation differs fundamentally from SOC estimation because the target changes slowly and is not uniquely defined. Capacity-based SOH, resistance growth, power capability, and application-specific performance loss require different reference tests [11,15,41,42]. A full charge or discharge under controlled temperature may provide a defensible laboratory label, but that label is rarely available during ordinary vehicle use [23,60]. The first question for every AI study is therefore not which model was used, but what health quantity was labelled, how it was measured, and whether the required input can be observed onboard [61].

Feature-based methods convert partial charge or discharge curves, incremental capacity analysis (ICA) peaks, impedance, resistance, temperature, or timing information into health indicators. Gaussian process regression (GPR) is attractive because it can combine nonlinear prediction with an explicit predictive variance, and it has been applied both to capacity trajectories and to charging-curve features [25,62,63]. Feature engineering can reduce model size and expose relationships that are easier to audit, but it transfers assumptions into preprocessing. Voltage-window availability, current regulation, smoothing bandwidth, interpolation grid, SOC alignment, and sensitivity to measurement resolution must therefore be reported. A feature that is stable during a controlled constant-current charge may be absent or distorted during opportunistic vehicle charging.

Deep networks reduce manual feature design by learning representations from cycling or partial-charging sequences. Compact neural screening and conventional ML pipelines show the range of possible data and deployment contexts [23,60]. Physics-informed and Bayesian approaches add structure or uncertainty [20,24], while refined charging-segment and Gaussian-process methods emphasize feature availability and confidence [25,31]. Their strongest contribution is the ability to fuse multiple weak degradation signatures, but the same flexibility makes leakage and domain dependence difficult to detect. For claims of generalization to previously unseen cells, cell-level holdout represents the minimum credible test. Cross-protocol, cross-temperature, cross-manufacturer, and cross-chemistry evaluations test additional transfer axes; success on one axis does not establish success on the others.

Early-life prediction and present SOH estimation should remain separate tasks. The large fast-charging dataset introduced by Severson and colleagues demonstrated that early-cycle voltage-curve features can predict eventual cycle life within a specific lithium iron phosphate (LFP)/graphite test campaign [64]. That result does not mean that the same features directly measure current capacity, nor does it establish transfer to another chemistry, charging policy, or vehicle duty cycle. Likewise, a model trained to reproduce a smooth capacity trajectory may have low interpolation error while failing to detect a knee point, an abrupt change in degradation mode, or a latent fault. The review therefore records the prediction horizon and EoL definition separately from the present-state error.

Additional sensing can increase information content. ML applied to electrochemical impedance spectra has separated degradation patterns and estimated capacity or RUL across cells, SOC levels, and temperatures [65]. Rapid impedance-based screening is also relevant to second-life assessment [60]. For continuous vehicle deployment, however, the measurement itself is part of the implementation burden. Excitation amplitude, frequency range, acquisition time, rest requirement, power-electronics capability, temperature compensation, and interference from balancing or vehicle operation determine whether the method belongs onboard, at a service station, or in a laboratory. High predictive accuracy from an information-rich diagnostic instrument should not be compared directly with estimation from standard BMS sensors.

Transfer and domain-adaptation methods address the shortage of labelled degradation data. Cross-domain DL has estimated SOH for target batteries without additional target degradation labels, and personalized transfer learning has been evaluated across many discharge protocols and battery datasets [66,67]. Recent work extends this direction to multiple chemistries and from cells to packs [68]. These studies move beyond single-cell interpolation, but practical claims still depend on how the source and target populations differ. The number of manufacturers, cells, packs, ageing paths, chemistries, and target-domain samples must be reported together with per-domain errors; a pooled average can conceal a poorly served battery group.

Field data narrow the gap between laboratory cycling and vehicle use but introduce a different labelling problem. On-road SOH models can exploit charging events, driving behavior, thermal history, and fleet-scale diversity, with cloud computation supporting feature extraction and model maintenance [34,69]. Yet vehicle data contain irregular sampling, software changes, sensor replacement, incomplete charging windows, unknown balancing activity, and maintenance interventions. Capacity labels may be inferred rather than measured and can share assumptions with the model being evaluated. A field study should therefore document label provenance, vehicle and pack holdout, data-quality filters, firmware versions, missing-data handling, and whether the test period occurs strictly after the training period.

A deployable SOH output should include uncertainty and a decision context. Bayesian DL [24] and GPR methods [25,62], including impedance-based models [65], offer predictive distributions whose calibration still requires testing on unseen cells and conditions. When uncertainty is high, the BMS may retain the last trusted estimate, enlarge operating margins, request a diagnostic event, or use a conservative model. The task-specific questions are whether the health target has a defensible reference test and whether the required measurements are available for the intended maintenance, warranty, charging, or power-limit decision.

SOC and SOH describe interacting time scales. Capacity fade and resistance growth alter the usable charge and voltage response used by SOC estimators, while many SOH methods use an SOC estimate or controlled charge window to interpret partial-charging data. Figure 2 shows these links and the assurance requirements shared by both functions.

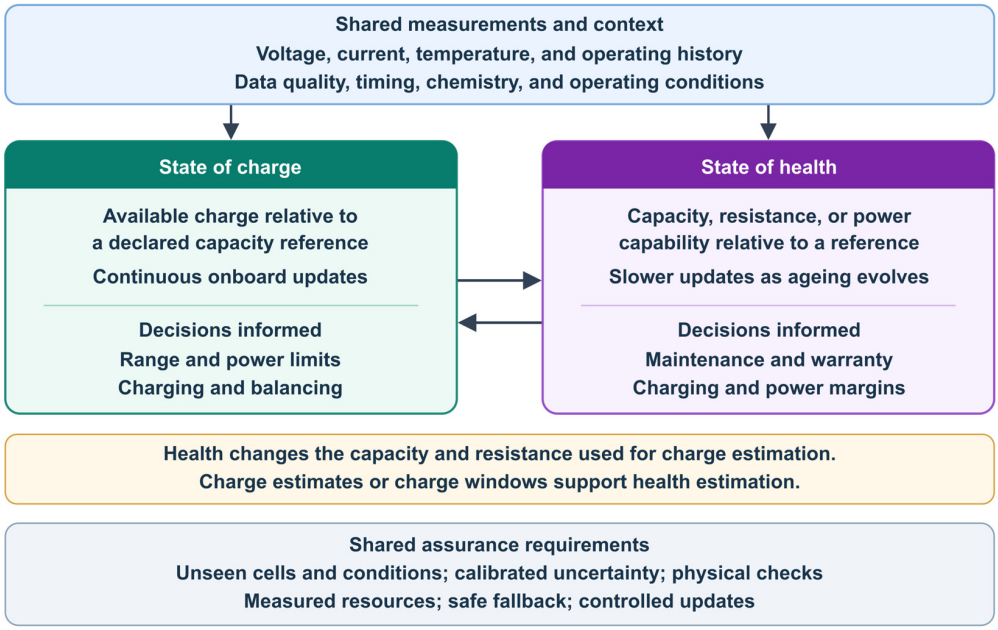


Figure 2. Coupled state-of-charge and state-of-health estimation, showing their targets, update time scales, decisions, and shared assurance requirements.

4.3. Remaining Useful Life and Degradation Trajectory

RUL prognosis should be separated from both present SOH estimation and retrospective curve fitting. A degradation-trajectory model predicts the future evolution of a specified health variable, whereas RUL converts that trajectory into time, cycles, energy throughput, or distance remaining before a declared EoL threshold is crossed. The target definition, forecast origin, horizon, threshold, future-duty assumption, and treatment of censored cells must therefore accompany every reported error. Reviews that combine SOC,

SOH, RUL, and knee-point results are useful maps of the field, but their metrics are not interchangeable [5,9].

Physics-informed and hybrid prognosis attempts to constrain extrapolation with degradation knowledge rather than relying on sequence correlation alone. Existing Bayesian physics-informed frameworks and physics-informed neural networks couple data with state or degradation equations [21,22]. A recent cycle-aware framework similarly embeds degradation structure in the prediction process [70]. Such constraints can reduce implausible trajectories, but they do not guarantee physical validity: the assumed mechanism, residual weighting, parameter identifiability, and ablation against an equally sized unconstrained model should be reported.

Probabilistic prognosis is especially important because uncertainty normally expands with the forecast horizon. Bayesian DL and Gaussian-process studies provide a basis for predictive distributions [24,62]. More recent work combines neural pseudo-observations with Bayesian filtering and explicitly targets trustworthy RUL uncertainty [71,72]. A narrow interval is not evidence of reliability unless its coverage and sharpness are tested on unseen cells; calibration should be stratified by horizon, temperature, protocol, chemistry, and proximity to the EoL threshold.

Model capacity is increasingly used to represent heterogeneous degradation modes. Mixture-of-experts and deformation mechanisms can allocate different submodels to different trajectory patterns [73], while augmentative generalization learning addresses uncertain working conditions [74]. These developments are promising, but benchmark gains may reflect more flexible interpolation rather than transferable prognosis. The review therefore distinguishes within-dataset holdout from tests that exclude entire cells, cycling policies, temperature regimes, manufacturers, or chemistries.

Knee-point prediction is a separate and safety-relevant challenge. A model can achieve a low average capacity error across most of life while missing the onset of accelerated degradation, because the pre-knee region dominates the sample count. Early-cycle features have demonstrated that future life can be predictable in a controlled fast-charging campaign [64], but the inferred relationship remains protocol- and chemistry-dependent. Knee location error, advance-warning time, false-alarm rate, and performance on cells without a pronounced knee should be reported in addition to trajectory MAE or RMSE.

Future operating conditions are part of the forecast, not an incidental input. A vehicle's temperature exposure, charging behavior, depth of discharge, power demand, storage periods, and software-imposed limits alter the degradation path. Prognosis based on a known laboratory protocol is conditional on that protocol; an onboard RUL value requires either a declared usage scenario, several scenario-conditioned forecasts, or an adaptive model that updates as duty changes. A recent multi-fidelity review reaches the same conclusion: transfer from laboratory cells to real packs requires validation across data fidelity, scale, and operating context [75].

Long-horizon prognosis can run during charging or service events, whereas an onboard advisory model must handle irregular partial histories, bounded memory, and delayed labels. RUL need not be updated at the SOC rate. Its added value should be judged by how a forecast changes inspection, charging, warranty, or retirement decisions and by the costs of acting early or late.

For RUL, the distinguishing evidence is a declared EoL threshold and causal forecast origin, realistic future-use assumptions, horizon-dependent calibration, and performance near knees and EoL. The shared requirements for independent testing and deployment appear in Tables 3 and 5, and Appendix A. Reconstruction using future samples is treated as offline characterization rather than prospective prognosis under unknown vehicle use.

4.4. Fault Diagnosis and Safety-Relevant Detection

Fault diagnosis is evaluated separately from state estimation because the operational objective is not a small average error but timely discrimination between normal variation and conditions that require intervention. Relevant faults include internal and external short circuits, abnormal self-discharge, overcharge or overdischarge, thermal and cooling-system faults, sensor bias or dropout, connection resistance, cell-voltage inconsistency, and failures of contactors or balancing circuits. Recent reviews broaden the taxonomy from laboratory signatures to prognosis and fault-tolerant control [76,77]. A lab-to-real-world safety review further emphasizes that detectability and response depend on where and how the fault evolves in the pack [78].

Class imbalance and label quality dominate this task. Fleet data contain many normal operating hours and comparatively few confirmed hazardous faults, while laboratory studies often induce faults under tightly controlled conditions. Supervised internal-short-circuit detection has shown that learned classifiers can use electrical and thermal signatures [17], and segmented-regression approaches provide another route to detecting abnormal voltage evolution [79]. However, a classifier trained on one imposed short resistance, sensor layout, or thermal boundary may not recognize an incipient field fault with a different mechanism.

Thermal diagnosis illustrates the pack-level observability and scalability problem. A physics-and-neural-network pack model can combine thermal expectations with learned residual patterns [80], whereas feature-augmented autoencoders can detect voltage anomalies without requiring a complete catalogue of labelled faults [81]. Neither approach should be judged by aggregate accuracy alone. The number and placement of sensors, cooling state, ambient temperature, cell imbalance, fault severity, onset profile, and the delay from physical initiation to detection determine whether an alarm is useful.

Hybrid and model-constrained methods are attractive because they compare measurements with an expected physical response. Model-constrained DL has been demonstrated for online diagnosis under stochastic conditions [32], and methodological reviews increasingly treat physics-ML coupling as an assurance mechanism rather than merely a way to improve fit [82]. The residual generator must nevertheless remain observable under real vehicle excitation, and its parameter drift, sensor dependence, and failure modes must be disclosed. If the learned and physical branches share the same biased sensor, their agreement is not independent confirmation.

Evaluation should use event-level rather than sample-level metrics. A million correctly classified normal samples can hide a missed hazardous event. Studies should report sensitivity and specificity by fault type and severity, false alarms per operating hour, time-to-detection, warning margin before a protection limit, localization accuracy, and the consequence of a missed or late alarm. Thresholds must be selected without using the final test events, and repeated windows from a single fault episode must not be split between training and testing.

Open-set behavior is equally important. Vehicle systems encounter combinations that were not represented during training, including simultaneous sensor drift and cooling degradation, service-induced configuration changes, and faults superimposed on aged or imbalanced cells. An anomaly score should therefore be calibrated against normal-domain diversity and paired with an out-of-distribution or residual check. The BMS response may range from logging and cloud review to power derating, charge interruption, isolation, or emergency shutdown; the model's role and authority must be stated for each response level.

Fleet collaboration can increase exposure to rare events, but it also creates privacy, heterogeneity, and governance constraints. A recent privacy-preserving framework uses heterogeneous charging-station data for collaborative fault warning across many vehicles [83]. This direction can improve coverage only if differences in battery design, firmware, sensor calibration, and charging context are retained in validation rather than averaged away. Se-

cure aggregation, provenance, update approval, rollback, and post-deployment monitoring become part of the diagnostic system.

The diagnostic result must connect to a verified response path. An alarm should operate alongside conventional electrical and thermal protection, plausibility checks, watchdogs, and a defined degraded mode. Event independence, detection delay by severity, and false alarms per operating hour are the task-specific criteria summarized in Table 3; offline classification alone does not establish safe intervention.

4.5. Thermal Management and Coupled Control

Thermal management couples estimation to physical actuation. A learned model may infer core temperature or heat generation, approximate a thermal model for predictive control, or recommend coolant flow, compressor power, preconditioning, and charging limits. Its benefit should be judged against an implementable controller using the same sensors, actuators, and thermal boundary conditions. Maximum temperature, spatial temperature spread, auxiliary energy, and constraint violations must be evaluated together; a smaller temperature error alone does not establish a better cooling decision [45,47].

Recent physical studies illustrate why the plant model matters. Anisotropic-conductivity measurements on a 114 Ah prismatic cell show that heat transport depends on direction, temperature, and swelling [84]. Liquid-immersion experiments on 280 Ah energy-storage modules investigate suppression of runaway propagation [85]. An interwound cooling-belt study examines how channel geometry and coolant conditions change temperature uniformity and thermal-hydraulic performance in cylindrical-cell modules [86]. These studies characterize physical constraints and cooling mechanisms; they do not validate an AI controller. The stationary-storage immersion results also require an explicit transfer assessment before use in an EV design.

For AI, these results imply that a surrogate trained with constant isotropic conductivity or a fixed cooling boundary may learn a response that changes with swelling, ageing, flow redistribution, or pack geometry. Validation should vary these properties and distinguish interpolated computational fluid dynamics predictions from predictions tested against independent physical measurements. Heat-transfer parameters and sensor placement should be documented together, because a surface-temperature fit can conceal an unobserved hot spot. Physics-based residuals can assist diagnosis: Naguib et al. combine a pack thermal model with a neural component and test induced faults on a 72-cell pack [80]. That experiment supports pack-level fault-detection evidence, while controller execution and intervention authority require separate verification.

Recent control research also shows why efficiency claims need a declared baseline. The publisher abstract for Xia et al. reports that deep deterministic policy gradient control reaches the battery temperature target earlier than model predictive control under a 40 °C New European Driving Cycle (NEDC) condition, with increased pump and compressor energy [87]. This supports a qualitative example of a response-time/energy trade-off; the record is not included in the quantitatively graded synthesis because its complete experimental and validation details were not accessible. A learned control policy should therefore report both thermal performance and total auxiliary energy under identical cabin and battery requirements, including the cost of sensing and computation.

Fast charging and ageing make the validation boundary stricter. Temperature-dependent resistance, uneven heat generation, coolant or sensor faults, and changing charge limits can invalidate a policy trained on fresh cells at one ambient condition. A credible experiment should use unseen duty cycles and temperatures, aged or deliberately imbalanced modules, charging transients, and failures such as flow restriction or sensor bias. It should disclose peak temperature, maximum cell-to-cell spread, time out-

side permitted limits, pump/compressor energy, degradation assumptions, and recovery after disturbances. Numerical values should remain tied to the tested chemistry, geometry, coolant, and protocol.

For control authority, the learning component should operate inside explicit thermal and electrical constraints. An independent supervisor should limit actuator commands, derate charging when cooling capacity is inadequate, and preserve protection during missed deadlines or model uncertainty. Training-time constraint penalties do not establish runtime constraint satisfaction. Promotion toward vehicle use requires measured controller execution and closed-loop pack or HiL tests, followed by monitored operation under a declared thermal envelope.

Table 3 consolidates the common comparison across the five application areas. Its task-specific evidence complements the methodological remedies in Table 5, deployment requirements in Sections 6–8, and reusable extraction checklist in Appendix A.

5. Critical Evaluation of Reported Performance

This section evaluates recurring weaknesses that can make an apparently accurate AI result non-transferable, overconfident, computationally impractical, or unsafe when placed inside a vehicle decision loop.

Table 4 aggregates representative primary results across estimation, prognosis, diagnosis, and embedded implementation. The studies were selected to show contrasting validation and execution contexts with retrievable experimental details, rather than to estimate the prevalence of any method or evidence category. Metrics are transcribed with their source scale and context; they are not pooled or ranked.

Table 4. Representative quantitative synthesis of primary AI-BMS studies. Metrics retain the original target and evaluation context. All table, figure, and section numbers in the last column refer to the cited original studies, not to this review.

Reference	AI Method	BMS Function	Dataset and Test	Reported Metrics	Evidence	Deployment Context
Mazzi et al. [6]	one-dimensional CNN; GRU; INT8 conversion	SOC	Panasonic UR18650ZY; two dynamic profiles	FP32 CNN RMSE 2.691%/2.355%; INT8 CNN 0.971 ms with STM32Cube.AI	E2 + E4-hardware	STM32F429ZI; errors are FP32, timing INT8; Tables 2, 5 and 6 in [6]
Pau and Aniballi [7]	Parallel temporal CNNs	SOC	LGHG2; held-out cycle sets 10/16 and 10/37	MSE 5.4/ 7.1×10^{-4}; MAPE 2.96%/3.26%	E2 + E4-hardware	SR5E1 and STM32 deployment; Tables 3, 4, 7–12 of [7]; no field validation
Giazitzis et al. [18]	Projected LSTM	SOC	Polimi LG18650HE4; cell 7 and selected drive cycles held out	MAE 0.0114; RMSE 0.0286 (source scale); model 4.0 kB	E3	0.391 ms on RTX 3060; MCU use proposed; Abstract, Sections 5 and 6 in [18]
Giazitzis et al. [18]	Projected CNN–GRU; ICA	SOH	Same cells; partial charging; cell 7 held out	MAE 0.0060; RMSE 0.0130 (source scale); model 3.4 kB	E3	1.607 ms on RTX 3060; no measured MCU timing; Table 4 of [18]
Rastegarpanah et al. [60]	Impedance features; optimized ANN	Capacity SOH	13 EoL Nissan Leaf modules; six-fold procedure	SOH RMSE (1.729 ± 0.147)%	E2	Laboratory screening; module origin does not establish onboard execution; Abstract, validation procedure
Severson et al. [64]	Early-cycle feature regression	Cycle life	124 LFP/graphite cells; 41 train/ 43 primary test/ 40 secondary test	Discharge model, secondary test: RMSE 173 cycles; mean percent error 8.6%	E3	First 100 cycles; EoL at 80% nominal capacity; Table 1 of [64], Methods

Table 4. *Cont.*

Reference	AI Method	BMS Function	Dataset and Test	Reported Metrics	Evidence	Deployment Context
Naha et al. [17]	Random forest; circuit-derived features	Internal short circuit	Smartphone cells; training and test batteries differ; abuse tests	At 5% discharge cutoff: 100% normal-in-normal; 98.45% fault-in-fault	E3	Inference tested on computer; phone records data; Discussion, Figures 4 and 5 in [17]
Naguib et al. [80]	Thermal physics plus DNN	Thermal fault diagnosis	72-cell air-cooled pack; drive cycles and induced faults	Fault types detected within about 45 min in reported tests	E2 + E4-pack	Eight temperature measurements; detection margin is fault-specific; no production timing claim
Chaoraingern and Numsomran [16]	TinyML FFNN; INT8	RUL	1100 mAh lithium-polymer; UAV experimental application	Experimental MAE 3.46 cycles; RMSE 3.75 cycles; INT8 latency 2 ms	E2 + E4-hardware	RP2040; 1.2 kB RAM/11 kB flash; UAV transfer evidence, not EV validation

When interpreting Table 4, it should be noted that percentage errors, dimensionless errors, cycle counts, and detection times refer to different targets and are therefore not directly comparable. In particular, the Giazitzis errors are retained on the published scale rather than converted without a documented normalization; their GPU timing is not MCU timing. The Mazzi FP32 errors and INT8 timing describe different model representations. Pau's MAPE is relative to SOC and is not an SOC error in percentage points. Rastegarpanah's six-fold procedure does not establish module-independent field generalization. Naha's computer-based inference is distinct from phone data collection. Chaoraingern and Numsomran report separate model-testing and application results; the row uses the application MAE/RMSE and the separately reported INT8 footprint. These distinctions prevent numerical aggregation from becoming an unsupported claim of comparative superiority.

Table 5 summarizes the recurring methodological failure modes that the following subsections use to evaluate reported AI-BMS performance.

Table 5. Recurring failure modes in reported AI-BMS performance and the required remedy.

Failure Mode	Why It Matters	Required Remedy
Data leakage	Related windows or cycles appear in training and test sets	Split by cell, pack, vehicle, or protocol; fit preprocessing only on training data
Domain shift	Chemistry, temperature, load, ageing, or firmware changes the distribution	Cross-domain tests, per-domain reporting, calibrated abstention and fallback
Weak labels	SOH, RUL, and fault labels may be delayed, inferred, or destructive	Report provenance, uncertainty, timing, and independence from model inputs
Metric narrowness	Average MAE, RMSE, or accuracy hides rare severe errors	Add maximum and subgroup error, calibration, delay, false alarms, and decision loss
Model complexity	Benchmark gains may violate controller budgets	Measure flash, peak RAM, precision, latency, energy, and post-compression performance
Nonstationarity	Ageing, service, and updates alter the data-generating process	Monitor drift, define revalidation, staged update, rollback, and retirement
Safety blindness	A correct prediction does not prove safe behavior under failure	Use independent monitors, physical limits, degraded modes, and fault-injection tests

5.1. Benchmark Discipline

Benchmark discipline begins with the experimental unit. Samples, windows, and cycles derived from one cell are correlated and cannot be treated as independent evidence. At minimum, the final split should hold out complete cells; stronger tests hold out packs,

vehicles, manufacturers, temperatures, protocols, or chemistries. Preprocessing parameters, feature selection, normalization, augmentation, hyperparameter tuning, and early stopping must be fitted without access to the final test group.

Every reported metric should be accompanied by the target definition and scale, battery chemistry and format, number of independent cells or packs, operating window, temperature and current ranges, charging protocol, sampling interval, input-window length, forecast horizon, and split strategy. For sequence models, the report should state whether windows overlap and whether future samples are used. For prognosis, it should state the EoL threshold and future-duty assumption. For diagnosis, it should report event counts, severities, and the unit over which false alarms are measured.

Baselines should represent the intended decision and implementation context. A complex neural estimator should be compared with Coulomb counting plus correction, an equivalent-circuit observer, or another implementable state estimator, not only with a weaker network. A fault detector should be compared with residual and threshold logic under the same sensor set. Recent comparisons of ML and Kalman-filter SOC estimators illustrate the value of testing conventional and learned alternatives under one protocol [88]. The benchmark should include parameter count, memory, timing, and post-compression accuracy whenever deployability is claimed.

The recurring failure modes and required remedies are summarized in the table above. The table is intentionally method-agnostic: leakage, domain shift, weak labels, metric narrowness, excessive complexity, nonstationarity, and safety blindness can invalidate a result regardless of whether the model is a neural network, Gaussian process, ensemble, or hybrid observer.

5.2. Uncertainty and Confidence

A safety-relevant BMS should provide a mechanism for identifying conditions under which its estimate may be unreliable. Predictive uncertainty concerns the target after accounting for measurement noise, model limits, and future variability; it is not the same as disagreement among repeated stochastic runs. Bayesian neural networks, Gaussian processes, ensembles, Monte Carlo dropout, conformal intervals, and observer residuals provide different approximations and assumptions. Trustworthy-model reviews increasingly place calibration and distribution shift alongside point accuracy [89].

Uncertainty must be evaluated on held-out cells and operating domains. Recommended measures include interval coverage and width, reliability diagrams, expected calibration error where appropriate, negative log-likelihood, and risk-coverage curves showing how error changes when the model abstains. Coverage should be reported by temperature, chemistry, ageing stage, and forecast horizon because acceptable average calibration can hide a subgroup with dangerous overconfidence. A production response also needs an explicit threshold and action: widen limits, revert to an observer, request a diagnostic event, or suppress an automated decision.

For BMS use, uncertainty should also distinguish aleatoric effects, such as measurement noise and operating variability, from epistemic effects caused by sparse data, unseen chemistries, or model-form uncertainty. The distinction matters because aleatoric uncertainty may justify wider operating margins, whereas epistemic uncertainty should often trigger abstention, fallback, or additional validation. Label uncertainty must be reported as well: an SOH label inferred from a partial charge, impedance feature, or smoothed capacity estimate is not equivalent to a controlled reference-test capacity.

5.3. Generalization, Leakage, and Domain Shift

Generalization claims should name the axis of novelty. An unseen window from a known cell tests interpolation; an unseen cell tests manufacturing variation; an unseen temperature or protocol tests condition shift; and an unseen chemistry, pack design, or fleet tests structural transfer. These levels should not be collapsed into a single test score. Acoustic state-estimation research provides a recent example of explicitly testing whether a promising signal generalizes beyond the development conditions [90].

Leakage can occur through overlapping windows, global normalization, capacity-derived features, label smoothing, repeated cycles, and model selection on the test set. Time-ordered splitting alone is insufficient when early and late windows from the same cell share a degradation identity. The review therefore records the grouping variable, preprocessing fit set, adaptation data, and whether target-domain labels were used. Fine-tuning on a target battery is reported as adaptation, not zero-shot transfer.

Domain-shift stress tests should perturb both inputs and context: current-sensor bias, voltage offset, missing samples, timing jitter, ambient-temperature change, cooling degradation, cell imbalance, charging-policy changes, and firmware-dependent preprocessing. Performance should be shown per domain and at the worst credible boundary. A pooled result may be insufficient when a small but safety-relevant subgroup carries most of the error.

5.4. Decision-Centered Evaluation and Failure Reporting

The value of an estimator or diagnostic model lies in the decision it changes. SOC error matters when it alters power limits, range, or protection margins; SOH and RUL error matter when they alter charging, warranty, maintenance, or retirement decisions; and fault scores matter through detection delay and unnecessary interventions. Evaluation should therefore add decision loss, threshold crossing error, warning margin, and the consequences of abstention or fallback to conventional point-error metrics.

Negative results and boundary failures should be retained as part of the evidence base rather than omitted from the deployment assessment. These include non-convergence from an incorrect initial state, loss of calibration near EoL, failure after quantization, timing overruns, sensitivity to sensor drift, inability to distinguish a fault from severe normal operation, and recovery after missing data. ISO/PAS 8800 places AI safety evidence within a lifecycle rather than treating benchmark accuracy as certification [8]. This lifecycle logic is operationalized later as an evidence-to-deployment validation ladder with explicit promotion gates from reproducibility to field monitoring.

6. Embedded Implementation Constraints

Embedded feasibility is a property of the complete sensing, preprocessing, inference, monitoring, and fallback chain, so model accuracy must be assessed together with resource budgets and worst-case behavior.

6.1. Compute, Memory, Timing, and Energy Budgets

A vehicle BMS is commonly distributed across cell-monitoring devices and one or more controllers with finite flash, RAM, processor time, communications bandwidth, thermal headroom, and power. The AI task competes with acquisition, balancing, diagnostics, communications, safety monitoring, and control. Parameter count is therefore an incomplete proxy. Reports should include executable and model storage, peak RAM including input buffers, arithmetic precision, measured average and worst-case latency, update rate, processor load, energy per inference or incremental power, toolchain, and the exact target controller.

Acceptable resource budgets are application-specific rather than universal. A slow SOH update during charging or parking can tolerate a different latency and energy budget

from a high-rate protection-adjacent estimate used during fast charging or regenerative braking. Reports should therefore state the control period, processor family, clock rate, flash and RAM allocation available to the AI function, arithmetic precision, measured worst-case execution time, and design margin under concurrent acquisition, communication, diagnostic, and balancing tasks.

Timing must be matched to the function. SOC and protection-related residuals may require frequent bounded updates, while SOH and RUL can often run during charging, rest, or scheduled background windows. Average desktop inference time does not establish real-time behavior on an automotive controller. Interrupt load, cache and memory contention, communication delays, cold start, watchdog limits, and simultaneous BMS tasks should be included in PIL or HIL measurements.

6.2. Compression, Quantization, and Compact Machine Learning

TinyML is a design discipline for fitting a complete and testable pipeline within a constrained target, not a label for any relatively small network [91]. Existing microcontroller SOC studies and hardware-accelerated implementations demonstrate the feasibility of compact inference [6,7]. Weng et al. demonstrate joint RUL and temperature prediction on a Raspberry Pi 4 edge device [92]. This supports edge execution, while transfer to an automotive MCU requires separate measurement. Each claim should nevertheless disclose feature extraction, retained history, runtime libraries, model representation, and the accuracy and calibration change measured after deployment.

Quantization-aware training, pruning, knowledge distillation, structured sparsity, lookup tables, fixed-point observers, and accelerators can reduce cost. Compression should be evaluated on the final target and across the original validation domains because a small change in average error can conceal a larger change near limits or under rare conditions. Deterministic execution, saturation behavior, numerical overflow, and reproducible compilation are as important as nominal throughput.

6.3. Onboard-Cloud Partitioning and Data Movement

Cloud support can host fleet analytics, long-horizon prognosis, model development, and update assessment, while onboard software retains functions needed for immediate safe operation. DT and cloud-side-end architectures illustrate this partition [48,49]. The boundary must specify which signals leave the vehicle, bandwidth and retention assumptions, behavior during disconnection, privacy controls, model-version compatibility, and which decisions remain possible without the cloud.

Data movement can cost more memory and energy than inference, particularly when long high-rate histories are buffered. Event-triggered summaries, feature extraction, and charging-session uploads can reduce the burden, but they may discard evidence needed for later fault analysis. Fleet collaboration for rare-fault warning adds value only when provenance and battery configuration are retained [83]. An onboard-cloud design should therefore be evaluated as an end-to-end information system, not as two independent models.

6.4. Runtime Assurance, Updates, and Fallback

Embedded execution does not remove the need for independent assurance. The AI output should be checked against physical bounds, rate limits, conservation relationships, observer residuals, and sensor plausibility. Runtime monitors must define what constitutes disagreement and what happens next: hold the last trusted value, revert to a conventional observer, enlarge safety margins, derate power, interrupt charging, or request service. The fallback path must be tested under the same injected faults and timing conditions as the learned component.

Updates introduce configuration and lifecycle risk. Training-data provenance, target hardware, compiler, calibration set, approval evidence, cryptographic integrity, rollback, and fleet monitoring should be versioned together. A prototype vehicle demonstration can show architectural feasibility [93], but production evidence additionally requires long-duration timing, environmental qualification, fault injection, and controlled update governance. The validation ladder developed in Section 8 therefore treats embedded execution as one promotion gate rather than the final proof of readiness.

Table 6 converts these implementation constraints into a minimum reporting checklist for embedded AI-BMS studies.

Table 6. Minimum embedded-deployment reporting checklist for an AI-enabled BMS.

Category	Required Fields	Evidence on Target Hardware	Release Criterion
Memory and compute	Parameters, executable/model storage, peak RAM, buffers, precision	Measured with final preprocessing and runtime	Fits allocated flash and peak RAM with declared design margin
Timing	Update rate, mean and worst-case latency, jitter, processor load	Concurrent acquisition, communications, diagnostics, and control	Worst-case execution time below task deadline with declared margin
Energy and thermal	Energy per inference or incremental power, accelerator use	Representative voltage, temperature, and duty	Incremental energy and heat remain inside the controller and pack thermal budget
Numerical behavior	Quantization, saturation, overflow, missing data, cold start	Boundary vectors and long-duration execution	Bounded outputs, no overflow, and reproducible compilation across release builds
Fault response	Domain check, watchdog, abstention, fallback, recovery	Injected sensor, timing, memory, model, and communication faults	Safe degraded operation, logged trigger, and verified recovery or service request
Lifecycle control	Model/data version, compatibility, signature, update, rollback	Interrupted and incompatible update tests	Signed, compatible release with tested rollback and traceable configuration record

7. Explainability, Safety, and Cybersecurity

Explainability, safety assurance, and cybersecurity are treated here as connected lifecycle obligations because an interpretable output is useful only when it supports a defined engineering decision, remains within a verified safety architecture, and cannot be corrupted through the data or update path.

7.1. Explainability as Engineering Evidence

The black-box problem is not simply that a neural network is mathematically complex. The engineering problem is which inputs or internal factors were influential in producing an output, whether the reason is stable under small perturbations, and whether the model relies on a physically meaningful signal or a spurious feature. Explainable-AI surveys provide a broad toolbox, while BMS-specific reviews increasingly connect explanations to state estimation and health management [33,94]. Post-hoc explanations can reveal correlations that deserve investigation, but they do not by themselves prove causal correctness, robustness, or safe behavior.

The explanation method should match the failure hypothesis. Feature attribution can identify dependence on voltage, current, temperature, or a narrow part of a charging curve; counterfactual analysis can test how much a measurement must change before the decision changes; surrogate models can summarize local behavior; and example-based explanations can expose reliance on a small subset of cells. Each method has its own approximation error.

Faithfulness to the original model, stability across repeated runs, sensitivity to baseline choice, and consistency with known electrochemical behavior should therefore be evaluated rather than assumed.

During development, explanations can help investigate leakage, sensor shortcuts, and dependence on particular cells or conditions. Divya et al. provide an illustrative battery-health example using simulated data [95]; it does not establish vehicle validation or a safety benefit. For engineering use, an explanation should be checked against a defined failure hypothesis and physical evidence, while independent limits and uncertainty determine the operational response.

Intrinsic constraints are often easier to assure than unrestricted post-hoc narratives. Monotonic relationships, bounded state transitions, conservation checks, sparse feature sets, interpretable residuals, and hybrid observers restrict what the model can express. Their value should be demonstrated by ablation and stress testing: the constraint must improve physical plausibility or failure detection without concealing a loss of accuracy in a relevant operating region.

7.2. Functional Safety and the Assurance Case

Functional safety starts from the vehicle hazard and safety goal, not from the learning algorithm. ISO 26262 structures the lifecycle for safety-related electrical and electronic systems [96], while ISO/PAS 8800 addresses additional evidence for road-vehicle functions that use AI [8]. Together they imply traceable requirements, controlled data and model configuration, verification at multiple integration levels, and production monitoring. Compliance cannot be inferred from low MAE, a high F1 score, or the presence of an explanation.

A defensible architecture separates the learned function from the mechanisms that keep the battery within a safe envelope. Sensor plausibility checks, voltage and temperature limits, an independent observer or residual, uncertainty thresholds, a runtime monitor, and a deterministic degraded mode provide layers with different failure assumptions. The AI component may improve estimation or early warning, but the safety argument must state which hazards it can influence, what authority it has, and which protection remains effective when it is wrong or unavailable.

A concrete assurance case for an AI-assisted SOC, SOH, or fault-warning function would normally include the claim that the learned output cannot by itself move the battery outside a safe electrical or thermal envelope. Supporting arguments would link hazards to safety goals, define the AI item boundary and operating design domain, identify data and model assumptions, justify performance and uncertainty requirements near protection limits, and show independent evidence from simulation, held-out data, fault injection, PIL testing, and HIL or pack tests. The unresolved residual risk is then controlled by limited authority, monitoring, and a deterministic fallback.

Verification should cover normal performance, boundary conditions, foreseeable misuse, sensor faults, resource exhaustion, distribution shift, and update failure. Requirements need measurable acceptance criteria for maximum error near protection limits, detection delay, confidence calibration, worst-case execution time, and recovery after fallback. Evidence should be linked to the exact dataset, preprocessing version, model binary, compiler, hardware, thresholds, and safety mechanism used in the tested configuration.

Broader AI governance frameworks can complement the automotive lifecycle. The National Institute of Standards and Technology (NIST) AI Risk Management Framework (AI RMF) organizes risk work around governance, mapping, measurement, and management [97], while the European Union (EU) AI Act establishes horizontal obligations whose applicability depends on the system's legal role and product context [98]. These sources are

useful for governance and post-market thinking, but they do not replace the vehicle-specific hazard analysis, technical safety concept, or type-approval obligations.

Table 7 maps the principal AI-related risks to engineering controls, verification evidence, and the residual decision that must remain explicit in a safety case.

Table 7. Safety-assurance mapping for AI-enabled BMS functions.

AI-Related Risk	Engineering Control	Verification Evidence	Residual Decision
Out-of-domain input	Domain detector, uncertainty threshold, physical bounds	Cross-condition stress tests and calibrated risk-coverage	Abstain, enlarge margin, or revert
Physically implausible output	Independent observer, conservation and rate checks	Fault injection and boundary-condition tests	Reject output and use fallback
Timing or resource overrun	Budget monitor, watchdog, bounded execution path	Worst-case timing under concurrent load	Skip AI update or enter degraded mode
Corrupted data or model	Authenticated data path, secure boot, signed model, protected calibration	Penetration, integrity, replay, and rollback tests	Isolate source and restore trusted version
Unsafe model update	Versioned evidence, staged rollout, compatibility and rollback	Interrupted update and shadow-fleet monitoring	Block, rollback, or retire release
Unexplained field drift	Logging, incident triage, controlled revalidation	Trend review by pack, firmware, climate, and use	Derate, service, or suspend model authority

7.3. Cybersecurity, Data Integrity, and Software Updates

AI expands the BMS attack surface through training data, calibration files, model binaries, update channels, cloud interfaces, diagnostic services, and the sensors on which inference depends. Adversarial manipulation need not resemble an image-classification attack: a current bias, replayed charging segment, altered timestamp, forged configuration identifier, or poisoned fleet label can shift SOC, SOH, RUL, or fault decisions while remaining electrically plausible. Charging-network security sources [99,100] are used here only to identify adjacent communication and update threats. They do not validate battery-state inference or prove that their network models fit a vehicle BMS controller.

ISO and SAE International (formerly the Society of Automotive Engineers) jointly publish ISO/SAE 21434, which provides a vehicle cybersecurity engineering lifecycle [101], and United Nations (UN) Regulation No. 155 requires a cybersecurity management system for relevant vehicle approvals [102]. For an AI-enabled BMS, the threat analysis should cover data provenance, authenticated measurement paths, secure boot, signed models, protected calibration, least-privilege diagnostics, communication segmentation, cloud access, logging, and key management. Security controls should be tested against the operational timing and memory constraints of the target controller.

Model updates require equal attention. UN Regulation No. 156 addresses software-update and software-update-management processes [103]. A governed update package should bind the model to its preprocessing, thresholds, compatibility matrix, safety evidence, and rollback image. The vehicle must reject incomplete or incompatible packages, preserve safe operation during an interrupted update, and retain enough version and event information to reproduce a field decision.

Cybersecurity and functional safety converge at runtime. A compromised model and an unintentionally drifted model can produce indistinguishable unsafe outputs at the system interface; therefore, independent monitors should respond to the observable consequences rather than relying solely on cause attribution. Logged interventions, security events, out-of-

distribution alarms, and rollback outcomes should feed a controlled incident-learning process without allowing unverified fleet data to modify a safety-relevant model automatically.

This convergence should be represented explicitly in the assurance argument. A model-integrity violation, a poisoned fleet label, a replayed charging segment, and an ordinary distribution shift can all manifest as the same unsafe estimate or missed warning. The runtime monitor should therefore check observable consequences, such as implausible state transitions, residual growth, inconsistent pack identifiers, impossible timestamps, and disagreement between learned and physical branches, while the cybersecurity case protects the assets and channels that could create those consequences.

8. Pathways to Practical Artificial Intelligence in Battery Management Systems

The practical pathways below are organized by the deployment constraint they solve and by the evidence needed to prevent a promising laboratory method from acquiring authority prematurely.

The double-edged character of AI can be made explicit as a trade-off test: each proposed learned function should identify the expected benefit, the new risk introduced, the evidence that the benefit survives realistic validation, and the maximum authority granted to the model if the evidence is incomplete. Accuracy, early warning, personalization, and fleet learning are the positive edge; opacity, data dependence, resource load, update risk, and cyber–physical attack surface are the negative edge. Practical deployment is justified only when the former is demonstrated under the same conditions in which the latter is controlled.

Table 8 summarizes the practical deployment pathways, the constraint each pathway addresses, and the evidence needed before it can justify promotion toward vehicle use.

Table 8. Deployment pathways, the constraints they address, and the evidence required before promotion.

Pathway	Problem Addressed	Evidence Required	Anchor References
TinyML and compression	Memory, latency, energy, and deterministic inference	Quantization, pruning, distillation, HIL measurement	[6,7,18,91,92]
Physics-informed learning	Poor extrapolation and physically implausible outputs	Physics residuals, constrained state transitions, hybrid observers and ablation tests	[19–22,70,82]
Uncertainty-aware models	Overconfident predictions under domain shift	Calibrated intervals, distribution shift detection and safe fallback thresholds	[23–25,71,72,89]
Explainable and interpretable models	Debugging, engineering review and assurance arguments	Intrinsic constraints, feature attribution audits and counterfactual stress tests	[33,34,94,95]
DTs and fleet learning	Sparse data and lifecycle adaptation	Data provenance, privacy, update governance and independent validation	[34,48,49,69,83]
Runtime safety monitors	Unknown unknowns and model degradation in the field	Residual checks, redundancy, watchdogs, and logged intervention events	[8,32,78–81,96,101–103]

8.1. Recommended Reference Architecture

The recommended architecture is a hybrid stack in which data quality, physical estimation, compact learning, runtime assurance, and actuation remain distinguishable. As introduced earlier in Figure 1, the learned component receives conditioned inputs and can refine an observer, estimate slow health variables, or generate an early-warning residual, but

its output passes through physical plausibility, uncertainty, and distribution-shift checks before it can influence a safety- or control-relevant BMS decision.

The sensor and context layer aligns timestamps, filters noise, checks range and rate limits, identifies the active pack and firmware configuration, and records missing or substituted data. The conventional layer maintains an equivalent-circuit, electrothermal, or rule-based representation and independent protection limits. The AI layer is deliberately narrow: compact inference is assigned only to tasks for which it provides a measured advantage, such as residual correction, transferable health indicators, or anomaly ranking.

The assurance layer evaluates disagreement among estimators and determines the appropriate system response. It compares learned outputs with state bounds, observer residuals, uncertainty, and the validated operating domain. The response is graded: accept, accept with enlarged margin, defer to the conventional estimate, request a diagnostic event, derate power, interrupt charging, or enter a deterministic safe state. This architecture supports physics-informed models and DTs [19,34], but it does not depend on any single model family.

Cloud services are kept outside time-critical local protection loops in the proposed architecture. They can aggregate fleet evidence, train candidate models, compare pack populations, and prepare signed releases, as demonstrated conceptually by cloud and cloud-side-end BMS architectures [48,49]. Every release should return through configuration control, target-hardware verification, and staged deployment. Measured edge inference [92] and a prototype vehicle architecture [93] illustrate different stages of implementation. The final preprocessing, precision, memory, timing, and decision authority must be documented for the configuration claimed.

8.2. Validation Ladder

The validation ladder defines five proposed deployment stages, from reproducible analysis to monitored vehicle operation. Each stage states the claim supported and the evidence required for the intended decision. Progression requires meeting predeclared acceptance criteria, rather than merely improving a benchmark score. Figure 3 summarizes the stages.

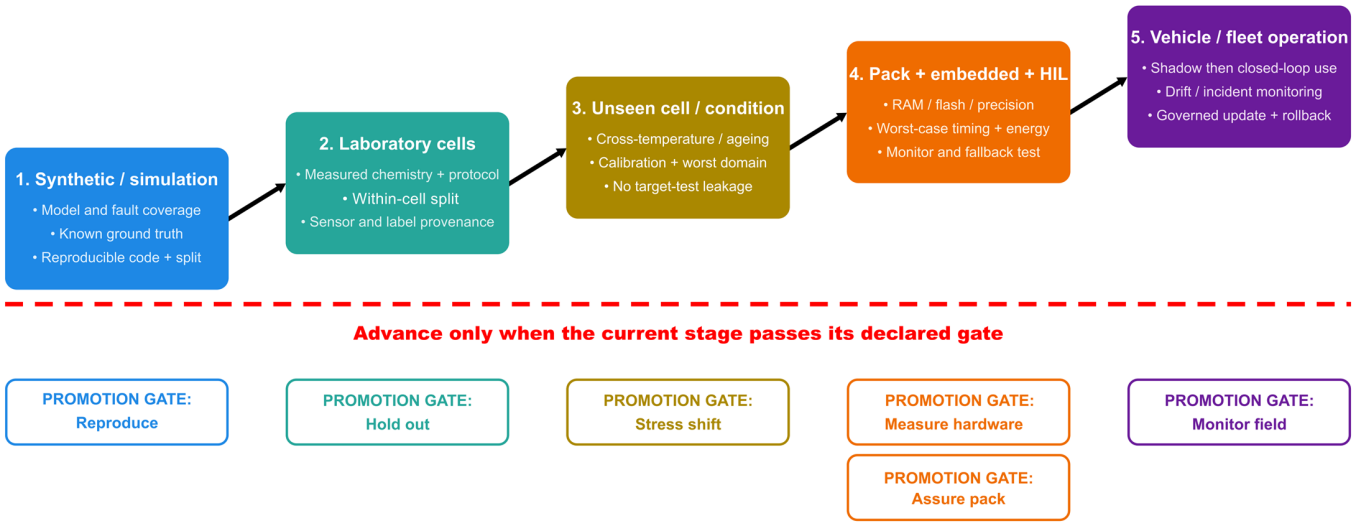


Figure 3. Evidence-to-deployment validation ladder for AI-enabled BMS. Promotion proceeds from reproducible simulation and laboratory evidence through independent cross-condition validation, measured embedded execution, pack-level assurance, and monitored vehicle or fleet operation.

Three terms are used consistently:

- Level (L1–L5): a proposed deployment state. E1–E3 describe data validation, and E4 records implementation or operational demonstrations; these descriptors do not automatically confer an L-level.
- Gate: the pass/fail decision for progression, based on the required evidence and predeclared acceptance criteria.
- Dimension: an aspect of readiness assessed by checklist items, including reproducibility, generalization, calibration, resource feasibility, safety behavior, and lifecycle governance. Failure or missing evidence on a required dimension blocks progression; strength elsewhere cannot compensate.

The five levels are:

1. L1—simulation and reproducibility. Simulated signals or synthetic faults. Evidence is internal consistency and reproducibility.
2. L2—controlled laboratory evidence. Measured data establish performance within the tested cell/protocol split; independent-cell or independent-condition transfer remains unproven.
3. L3—cross-condition and cross-cell evidence. Entire cells, temperatures, protocols, or chemistries excluded from training. Evidence is transfer beyond the calibration domain.
4. L4—pack and embedded verification. Independent-transfer evidence is combined with measured execution on the intended controller or a documented target-equivalent platform and module/pack tests. The final pipeline must meet memory, worst-case timing, and energy budgets under concurrent tasks, while PIL/HIL or pack tests verify monitors, faults, and fallback.
5. L5—vehicle and fleet evidence. Monitored vehicle or fleet operation, first in shadow mode, then bounded closed-loop authority. Evidence is real-world robustness and lifecycle governance.

Six checks support promotion. The L4 assessment includes both measured embedded execution and pack-level assurance because these address distinct failure modes and require complementary tests; they may be investigated together in an integrated experiment. A check contributes to the claimed level only when the relevant conditions and acceptance criteria are documented.

1. Reproducibility. Re-run the method with the original dataset, split, preprocessing, and software. Pass if the result is reproduced within tolerance or discrepancies are explained.
2. Independent holdout. Repeat with complete cell- or pack-level holdout. Pass if the method is evaluated on units excluded from training, preprocessing fit, tuning, and model selection, with per-group results reported.
3. Cross-condition stress. Test across temperature, C-rate, sensor bias, missing data, timing jitter, chemistry, ageing, and protocols. Pass if the method stays within declared accuracy and calibration bounds or correctly abstains out of domain.
4. Embedded execution. Deploy the final preprocessing and model on representative hardware. Measure flash, peak RAM, precision, average and worst-case latency, processor load, and energy under concurrent BMS tasks. Pass if the method fits its budget with margin, meets deadlines, and shows no unacceptable loss after compression.
5. Pack-level assurance. Inject sensor, communication, timing, memory, model, and thermal faults through SIL/PIL/HIL and controlled pack tests. Pass if monitor-triggered abstention, fallback, recovery, and safe actuation are demonstrated when the model is wrong, uncertain, or unavailable.
6. Monitored fleet operation. Begin with shadow deployment, then bounded closed-loop authority with defined monitoring, update approval, rollback, and retirement

criteria. Pass if the method performs acceptably under real diversity and its lifecycle is governed.

Promotion depends on the weakest unresolved requirement for the intended decision. A method that generalizes but misses its controller deadline cannot pass the embedded gate. A fast method that is overconfident under chemistry shift fails the corresponding robustness/calibration requirement. The same model may therefore be supported for advisory maintenance while lacking evidence for power-limit control. Table 9 proposes minimum validation states and function-specific evidence; these are engineering recommendations, not empirically validated thresholds.

Table 9. Minimum validation level and function-specific evidence by BMS function.

BMS Function	Minimum Level for Advisory Use	Minimum Level for Control or Protection Use	Function-Specific Evidence Required
SOC estimation	L3	L4	Causal update rate, bounded drift, recovery from initialization, calibration near voltage limits
SOH estimation	L3	L4	Defined health target, partial-charge input availability, cell- or pack-level holdout, cross-condition transfer
RUL prognosis	L3	L4	Declared threshold and horizon, future-duty assumption, horizon-dependent calibration, knee-point detection
Fault diagnosis	L3	L4	Event-level metrics, severity-stratified detection delay, false alarms per operating hour, response path
Thermal-state estimation and advisory	L3	L4	Sensor placement, hot-spot observability, cross-condition transfer, and coupling with electrical states
Charging and balancing control	L3	L5	Closed-loop stability, constraint satisfaction, cell-consistency behavior, fallback under actuator failure
Thermal management control	L4	L5	L4: closed-loop pack testing and actuator/fallback checks. L5: monitored vehicle actuation within declared thermal and energy limits

The minima in Table 9 depend on the intended decision. Advisory estimates generally require L3 evidence of the relevant transfer or forecast claim. Direct influence on control or protection additionally requires L4 embedded and pack verification. Thermal, charging, and balancing controllers can be tested at L4, but monitored vehicle actuation is assigned L5 because repeated actions can accumulate thermal stress or cell imbalance. These proposed minima complement hazard analysis and do not imply certification.

The ladder is not a substitute for a safety case. Passing all gates provides evidence of reproducibility, generalization, resource feasibility, and safe behavior under tested conditions. It does not by itself satisfy ISO 26262, ISO/PAS 8800, ISO/SAE 21434, or UN Regulations Nos. 155 and 156 obligations. Those require a traceable assurance argument linking requirements, data, model configuration, tests, monitors, fallback, field metrics, and change approval to the vehicle hazard analysis. The ladder supplies the technical evidence that such an argument consumes; it does not replace the argument. ISO/PAS 8800 and broader risk-management guidance reinforce this lifecycle view [8,97].

9. Industrial Evidence and Research Roadmap

Industrial evidence is evaluated separately from academic benchmarks because production programs expose AI-enabled battery functions to fleet diversity, configuration control, service processes, regulation, and commercial confidentiality, while public disclosures rarely reveal enough detail to reproduce a safety-relevant algorithm.

9.1. What Public Industrial Evidence Can Establish

Public company reports, product announcements, patents, supplier material, regulatory filings, and recall documents can establish system direction, operating scale, chemistry diversity, fleet exposure, and the existence of monitoring or update infrastructure. They often do not provide sufficient information to establish the model architecture, training population, validation strategy, confidence calibration, computational budget, safety classification, or fallback thresholds. The review therefore treats industrial material as evidence of context and deployment practice, not as proof that a proprietary AI estimator is more accurate or safer than a published baseline.

Tesla's public impact reporting provides fleet-level battery-retention summaries [104]. Such aggregate data are valuable because they reflect real vehicles, climates, charging behavior, and long-duration use, but they are not a validation report for a specific BMS software and configuration governance, and they do not reveal proprietary inference logic. Claims about a company's AI model, safety monitor, or embedded compute should therefore be made only when primary technical evidence supports them.

Peer-reviewed field studies provide a stronger bridge. On-road SOH estimation and privacy-preserving fleet fault warning demonstrate that vehicle-scale data can support transfer, personalization, and rare-event learning [69,83]. Even these studies require scrutiny of labels, vehicle holdout, firmware changes, privacy, and intervention outcomes. The strongest industrial evidence would combine a preregistered validation protocol, representative pack populations, independent audit, hardware measurements, and monitored operational decisions.

Industrial readiness should therefore be reported as an evidence matrix: function and decision authority; validated chemistries and pack variants; operating-domain coverage; hardware and timing; safety monitors and fallback; cybersecurity and update controls; field exposure; incident and rollback history; and evidence availability. Confidential details may remain protected, but the acceptance criteria and assurance logic must be reviewable by the organizations responsible for vehicle safety and approval.

9.2. Staged Research and Deployment Roadmap

The near-term priority is benchmark discipline. Public datasets should preserve cell identity, protocol, temperature, ageing path, sensor quality, and pack hierarchy, while reference tasks should clearly specify the experimental unit and the holdout strategy appropriate to the claimed form of generalization, and should include conventional baseline methods. Results should report calibration, worst-case and subgroup errors, fault-event metrics, and embedded resource requirements rather than relying solely on a single average performance score.

The second priority is compact hybrid intelligence. Research should target the smallest model that provides a decision-relevant improvement over a physical observer, with explicit uncertainty, domain checks, and an independent fallback. Physics-informed learning, transfer, TinyML, and DTs should be evaluated as responses to identifiable deployment constraints rather than as ends in themselves [75,82,92].

The third priority is cross-scale and cross-fleet validation. Candidate methods should progress from unseen cells to modules and packs, then to processor and HIL tests, controlled vehicles, shadow fleets, and bounded closed-loop authority. Data from multiple

manufacturers and chemistries are needed to distinguish transferable degradation signatures from pack-specific shortcuts. Privacy-preserving collaboration can increase rare-fault coverage, provided that provenance and configuration are retained [83].

The fourth priority is an explicit assurance case. Safety, cybersecurity, software-update, and AI-governance evidence should be linked to the same versioned model and target configuration [8,96,101]. The case should show how requirements, datasets, tests, explanations, monitors, fallback, field metrics, and change approval support the vehicle hazard analysis. Post-market monitoring should have predefined escalation, rollback, and retirement criteria.

The final objective is not an autonomous black-box BMS, but a monitored learning subsystem whose benefit is measurable and whose authority is bounded. Progress should be judged by reproducibility, transfer, calibrated uncertainty, embedded feasibility, safety behavior, and operational evidence. This roadmap turns the double-edged character of AI into an engineering decision: deploy the learned function only where its verified benefit exceeds the additional data, compute, assurance, and lifecycle risk it introduces.

10. Conclusions

This critical review assessed when AI results support a defined vehicle BMS use. Its contribution is to connect reported performance with validation independence, execution context, decision authority, and assurance. The evidence supports selective use of compact learned components within monitored BMS architectures; it does not establish one generally superior algorithm or a universal readiness threshold.

For RQ1, documented benefits include nonlinear state estimation, extraction of degradation indicators, conditional lifetime prediction, and fault-pattern detection. Their value depends on comparison with feasible conventional methods under matched inputs and operating conditions. Thermal control additionally requires joint evaluation of temperature limits, uniformity, response, and auxiliary energy.

For RQ2, transfer remains bounded by the cells, chemistries, protocols, ageing histories, and conditions tested. Independent cell, pack, or event holdout is more informative than randomly divided correlated windows. Table 4 illustrates why numerical results from different targets and splits cannot support a single performance ranking.

For RQ3, published MCU experiments demonstrate that compact inference is feasible for selected tasks, while GPU timing and model size alone leave embedded performance unresolved. The complete sensing, preprocessing, inference, monitoring, and fallback chain requires measurement on the intended controller.

For RQ4, uncertainty, explanation, and physical constraints contribute evidence only when tested against a defined failure hypothesis and response. Independent protection, bounded authority, fault injection, and controlled updates remain necessary parts of a safety argument.

For RQ5, the main remaining gaps are representative pack and fleet data, reproducible comparisons, calibrated behavior under shift, and demonstrated integration over the software lifecycle. The proposed checklist and deployment ladder organize these gaps but require independent validation. Future work should test their reproducibility alongside shared benchmarks, HIL evaluation, and monitored field studies.

The review is limited by targeted source selection, heterogeneous reporting, incomplete access to some recent studies, and sparse public production evidence. Its representative quantitative synthesis supports critical interpretation rather than meta-analysis. Deployment should follow demonstrated benefit and verified constraints for the intended decision.

Author Contributions: Conceptualization, L.S.; methodology, C.P.; investigation, C.P., N.K. and L.S.; resources, N.K.; writing—original draft preparation, C.P., N.K. and L.S.; writing—review and editing, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: During preparation of this manuscript, the authors used Microsoft Copilot and Grammarly for language editing, manuscript polishing, reference checking, and consistency review. The authors critically reviewed and edited all outputs, verified the cited evidence, and took full responsibility for the final content.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BMS	Battery Management System
DL	Deep Learning
DT	Digital Twin
EoL	End of Life
EV	Electric Vehicle
FP32	32-bit Floating Point
GPU	Graphics Processing Unit
INT8	8-bit integer
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCU	Microcontroller Unit
ML	Machine Learning
MSE	Mean Squared Error
RMSE	Root Mean Square Error
RUL	Remaining Useful Life
SOC	State of Charge
SOE	State of Energy
SOH	State of Health
SOP	State of Power
UAV	Unmanned Aerial Vehicle

Appendix A. Evidence Classification and Critical Appraisal Checklist

The Appendix provides a reusable record for the E1–E3 evidence classes and E4 implementation qualifiers and the separate methodological appraisal. The checklist is proposed for this review; independent validation, inter-rater reliability, and predictive validity have not been established.

For each assessment, record the reference and version, task and decision, source page/table/figure, extracted result and units, validation unit, execution context, category and qualifier, rationale, and unresolved questions. Use Yes, No, Unclear, or Not applicable for each checklist item, and retain the source passage or location. Do not convert the answers into a summed quality score.

Table A1 provides the extraction checklist. Assign E1, E2, or E3 from the demonstrated data validation, and record any E4-hardware, E4-pack, or E4-field qualifier separately. If a decisive detail is unavailable, record the class or qualifier as unassigned or provisional and identify the missing evidence. Physical cooling studies inform the thermal boundary but receive no AI-model evidence class.

Table A1. Category-specific evidence checklist.

Category	Mandatory Evidence and Questions	Record and Limit
E1 Simulation	Identify simulator, equations, parameter sources, synthetic fault process, boundary conditions, and test scenarios. Is a matching experimental test absent?	Record simulator and fault assumptions; no empirical or hardware-validation claim
E2 Laboratory	Identify experimental cells/modules/packs, protocol, labels, input window, split units, and metrics. Is testing limited to a within-cell or within-protocol split without demonstrated independent-cell or independent-condition transfer?	Record independent sample count and possible shared-unit leakage; controlled tests alone do not show transfer
E3 Transfer	Name units or conditions excluded from training, preprocessing fit, tuning, and selection. Document target adaptation, group-wise errors, and the external test source. Is the claimed transfer axis actually held out?	Record the specific axis and residual bias concerns; cell holdout does not prove chemistry or fleet transfer
E4 Implementation/ operation qualifier	Identify measured target or target-equivalent execution, or module/pack/field demonstration. Record preprocessing, precision, memory, timing, concurrent workload, use authority, and missing information. PIL/HIL supports E4-hardware only if the stated target is in the loop	Record E4-hardware, E4-pack, or E4-field separately from E1–E3. A demonstration does not establish all L4/L5 acceptance criteria

Table A2 lists the appraisal domains recorded independently of category. An E4-hardware study can still have unresolved cell-independent validation, and an E3 study can still have a serious analysis limitation. Record these as separate judgments. For mixed simulation/experimental papers, record which result each evidence category applies to rather than extending one successful experiment to every claim.

Table A2. Independent methodological appraisal domains.

Domain	Questions to Record as Yes/No/Unclear/Not Applicable	Consequence for Interpretation
Target and labels	Is the target defined, with units, reference test, threshold, forecast origin, and label provenance?	Unclear labels or targets limit any accuracy claim.
Data provenance and coverage	Are chemistry, format, units, protocols, temperatures, ageing, fault generation, and population documented?	Restrict the supported operating domain to observed coverage.
Independence and leakage	Are cell/pack/event identities separated where required? Are preprocessing, tuning, adaptation, and overlapping windows controlled?	Record leakage concerns separately from the evidence category.
Baselines and analysis	Are feasible conventional comparators, target scale, denominators, repeat variation, and subgroup or worst-case results given?	Do not infer superiority from unmatched datasets or metrics.
Uncertainty and robustness	Are intervals or abstention thresholds evaluated on independent data and relevant shifts? Are failure cases retained?	Predictive variance alone does not establish calibration.
Execution	Is the final pipeline run on the stated target? Are precision, buffering, peak memory, latency, concurrency, and energy disclosed?	Separate measured execution from desktop/GPU timing or projected capacity.
Safety response	Are authority, monitors, fallback, injected faults, actuator limits, and recovery demonstrated for the intended decision?	Detection or estimation accuracy alone does not establish safe actuation.
Transparency and reuse	Are data/code availability, source locations, versions, limitations, and adaptation requirements reported?	Mark missing information explicitly; do not interpret absence of reporting as a tested failure.

The observed E categories do not map one-to-one onto the L1–L5 deployment ladder. For example, an E4-pack fault detector can provide realistic pack measurements while still lacking the controller timing and fallback tests needed for L4 promotion. The minimum lev-

els proposed in Table 9 are engineering recommendations dependent on decision authority, not thresholds empirically validated against incidents or certification outcomes.

Future validation should use a prespecified manual and blinded independent assessments of a diverse sample of primary BMS studies, report agreement and adjudication, and examine whether individual checklist items predict replication, transfer, or successful embedded integration. The current paper reports no such validation study.

References

1. Wang, Y.; Tian, J.; Sun, Z.; Wang, L.; Xu, R.; Li, M.; Chen, Z. A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renew. Sustain. Energy Rev.* **2020**, *131*, 110015. [CrossRef]
2. Hu, X.; Feng, F.; Liu, K.; Zhang, L.; Xie, J.; Liu, B. State estimation for advanced battery management: Key challenges and future trends. *Renew. Sustain. Energy Rev.* **2019**, *114*, 109334. [CrossRef]
3. Nozarijouybari, Z.; Fathy, H.K. Machine learning for battery systems applications: Progress, challenges, and opportunities. *J. Power Sources* **2024**, *601*, 234272. [CrossRef]
4. Kurucan, M.; Özbaltan, M.; Yetgin, Z.; Alkaya, A. Applications of artificial neural network based battery management systems: A literature review. *Renew. Sustain. Energy Rev.* **2024**, *192*, 114262. [CrossRef]
5. Nazim, M.S.; Chakma, A.; Joha, M.I.; Alam, S.S.; Rahman, M.M.; Umam, M.K.S.; Jang, Y.M. Artificial intelligence for estimating State of Health and Remaining Useful Life of EV batteries: A systematic review. *ICT Express* **2025**, *11*, 769–789. [CrossRef]
6. Mazzi, Y.; Sassi, H.B.; Gaga, A.; Errahimi, F. State of charge estimation of an electric vehicle's battery using tiny neural network embedded on small microcontroller units. *Int. J. Energy Res.* **2022**, *46*, 8102–8119. [CrossRef]
7. Pau, D.P.; Aniballi, A. Tiny machine learning battery state-of-charge estimation hardware accelerated. *Appl. Sci.* **2024**, *14*, 6240. [CrossRef]
8. *ISO/PAS 8800:2024*; Road Vehicles—Safety and Artificial Intelligence (AI). ISO: Geneva, Switzerland, 2024.
9. Shah, A.; Shah, K.; Shah, C.; Shah, M. State of charge, remaining useful life and knee point estimation based on artificial intelligence and machine learning in lithium-ion EV batteries: A comprehensive review. *Renew. Energy Focus* **2022**, *42*, 146–164. [CrossRef]
10. Waag, W.; Fleischer, C.; Sauer, D.U. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *J. Power Sources* **2014**, *258*, 321–339. [CrossRef]
11. Berecibar, M.; Gandiaga, I.; Villarreal, I.; Omar, N.; Van Mierlo, J.; Van den Bossche, P. Critical review of state of health estimation methods of Li-ion batteries for real applications. *Renew. Sustain. Energy Rev.* **2016**, *56*, 572–587. [CrossRef]
12. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Mohamed, A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renew. Sustain. Energy Rev.* **2017**, *78*, 834–854. [CrossRef]
13. Plett, G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background. *J. Power Sources* **2004**, *134*, 252–261. [CrossRef]
14. Hu, X.; Li, S.; Peng, H. A comparative study of equivalent circuit models for Li-ion batteries. *J. Power Sources* **2012**, *198*, 359–367. [CrossRef]
15. Barre, A.; Deguilhem, B.; Grolleau, S.; Gerard, M.; Suard, F.; Riu, D. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *J. Power Sources* **2013**, *241*, 680–689. [CrossRef]
16. Chaoraingern, J.; Numsomran, A. Embedded sensor data fusion and TinyML for real-time remaining useful life estimation of UAV Li polymer batteries. *Sensors* **2025**, *25*, 3810. [CrossRef] [PubMed]
17. Naha, A.; Khandelwal, A.; Agarwal, S.; Tagade, P.; Hariharan, K.S.; Kaushik, A.; Yadu, A.; Kolake, S.M.; Han, S.; Oh, B. Internal short circuit detection in Li-ion batteries using supervised machine learning. *Sci. Rep.* **2020**, *10*, 1301. [CrossRef] [PubMed]
18. Giazitzis, S.; Ferrari, J.; Woss, A.; Badha, S.; Rosetti, F.; Ogliari, E. Embedded strategy for battery module states estimation using tiny machine learning models. *J. Energy Storage* **2026**, *152*, 120673. [CrossRef]
19. Wang, F.; Zhai, Z.; Zhao, Z.; Di, Y.; Chen, X.-F. Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis. *Nat. Commun.* **2024**, *15*, 4332. [CrossRef] [PubMed]
20. Hofmann, T.; Hamar, J.; Rogge, M.; Zoerr, C.; Erhard, S. Physics-informed neural networks for state of health estimation in lithium-ion batteries. *J. Electrochem. Soc.* **2023**, *170*, 090524. [CrossRef]
21. Nascimento, R.G.; Viana, F.A.C.; Corbetta, M.; Kulkarni, C.S. A framework for Li-ion battery prognosis based on hybrid Bayesian physics-informed neural networks. *Sci. Rep.* **2023**, *13*, 13856. [CrossRef] [PubMed]
22. Wen, P.; Ye, Z.S.; Li, Y.; Chen, S.; Xie, P.; Zhao, S. Physics-informed neural networks for prognostics and health management of lithium-ion batteries. *IEEE Trans. Intell. Veh.* **2024**, *9*, 2276–2289. [CrossRef]
23. Roman, D.; Saxena, S.; Robu, V.; Pecht, M.; Flynn, D. Machine learning pipeline for battery state-of-health estimation. *Nat. Mach. Intell.* **2021**, *3*, 447–456. [CrossRef]

24. Ke, Y.; Peng, W. A Bayesian deep learning pipeline for lithium-ion battery SOH estimation with uncertainty quantification. *Qual. Reliab. Eng. Int.* **2024**, *40*, 406–427. [CrossRef]
25. Pohlmann, S.; Mashayekh, A.; Stroebl, F.; Karnehm, D.; Kuder, M.; Neve, A.; Weyh, T. State-of-health prediction of lithium-ion batteries based on a low dimensional Gaussian Process Regression. *J. Energy Storage* **2024**, *88*, 111649. [CrossRef]
26. Collins, G.S.; Moons, K.G.M.; Dhiman, P.; Riley, R.D.; Beam, A.L.; Van Calster, B.; Ghassemi, M.; Liu, X.; Reitsma, J.B.; van Smeden, M.; et al. TRIPOD+AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ* **2024**, *385*, e078378. [CrossRef] [PubMed]
27. Moons, K.G.M.; Damen, J.A.A.; Kaul, T.; Hooft, L.; Andaur Navarro, C.; Dhiman, P.; Beam, A.L.; Van Calster, B.; Celi, L.A.; Denaxas, S.; et al. PROBAST+AI: An updated quality, risk of bias, and applicability assessment tool for prediction models using regression or artificial intelligence methods. *BMJ* **2025**, *388*, e082505. [CrossRef] [PubMed]
28. Shea, B.J.; Reeves, B.C.; Wells, G.; Thuku, M.; Hamel, C.; Moran, J.; Moher, D.; Tugwell, P.; Welch, V.; Kristjansson, E.; et al. AMSTAR 2: A critical appraisal tool for systematic reviews that include randomised or non-randomised studies of healthcare interventions, or both. *BMJ* **2017**, *358*, j4008. [CrossRef] [PubMed]
29. Whiting, P.F.; Savović, J.; Higgins, J.P.T.; Caldwell, D.M.; Reeves, B.C.; Shea, B.; Davies, P.; Kleijnen, J.; Churchill, R.; Group, R. ROBIS: A new tool to assess risk of bias in systematic reviews was developed. *J. Clin. Epidemiol.* **2016**, *69*, 225–234. [CrossRef] [PubMed]
30. Shea, B.J.; Grimshaw, J.M.; Wells, G.A.; Boers, M.; Andersson, N.; Hamel, C.; Porter, A.C.; Tugwell, P.; Moher, D.; Bouter, L.M. Development of AMSTAR: A measurement tool to assess the methodological quality of systematic reviews. *BMC Med. Res. Methodol.* **2007**, *7*, 10. [CrossRef] [PubMed]
31. Zheng, K.; Meng, J.; Yang, Z.; Zhou, F.; Yang, K.; Song, Z. Refined lithium-ion battery state of health estimation with charging segment adjustment. *Appl. Energy* **2024**, *375*, 124077. [CrossRef]
32. Cao, R.; Zhang, Z.; Shi, R.; Lu, J.; Zheng, Y.; Sun, Y.; Liu, X.; Yang, S. Model-constrained deep learning for online fault diagnosis in Li-ion batteries over stochastic conditions. *Nat. Commun.* **2025**, *16*, 1651. [CrossRef] [PubMed]
33. Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Rana, O.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Comput. Surv.* **2022**, *55*, 194. [CrossRef]
34. Njoku, J.N.; Nwakanma, C.I.; Kim, D.-S. Explainable data-driven digital twins for predicting battery states in electric vehicles. *IEEE Access* **2024**, *12*, 83480–83501. [CrossRef]
35. Kertész, N.; Szabó, L. Advances and future trends in battery management systems. *Eng. Proc.* **2024**, *79*, 66. [CrossRef]
36. Lelie, M.; Braun, T.; Knips, M.; Nordmann, H.; Ringbeck, F.; Zappen, H.; Sauer, D.U. Battery management system hardware concepts: An overview. *Appl. Sci.* **2018**, *8*, 534. [CrossRef]
37. Liu, K.; Li, K.; Peng, Q.; Zhang, C. A brief review on key technologies in the battery management system of electric vehicles. *Front. Mech. Eng.* **2019**, *14*, 47–64. [CrossRef]
38. Han, X.; Lu, L.; Zheng, Y. A review on the key issues of the lithium-ion battery degradation among the whole life cycle. *eTransportation* **2019**, *1*, 100005. [CrossRef]
39. Xiong, R.; Cao, J.; Yu, Q.; He, H.; Sun, F. Critical review on the battery state of charge estimation methods for electric vehicles. *IEEE Access* **2018**, *6*, 1832–1843. [CrossRef]
40. Farmann, A.; Sauer, D.U. A comprehensive review of on-board State-of-Available-Power prediction techniques for lithium-ion batteries in electric vehicles. *J. Power Sources* **2016**, *329*, 123–137. [CrossRef]
41. Farmann, A.; Waag, W.; Marongiu, A.; Sauer, D.U. Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles. *J. Power Sources* **2015**, *281*, 114–130. [CrossRef]
42. Birkl, C.R.; Roberts, M.R.; McTurk, E.; Bruce, P.G.; Howey, D.A. Degradation diagnostics for lithium ion cells. *J. Power Sources* **2017**, *341*, 373–386. [CrossRef]
43. Dubarry, M.; Vuillaume, N.; Liaw, B.Y. From single cell model to battery pack simulation for Li-ion batteries. *J. Power Sources* **2009**, *186*, 500–507. [CrossRef]
44. Hoque, M.M.; Hannan, M.A.; Mohamed, A.; Ayob, A. Battery charge equalization controller in electric vehicle applications: A review. *Renew. Sustain. Energy Rev.* **2017**, *75*, 1363–1385. [CrossRef]
45. Bandhauer, T.M.; Garimella, S.; Fuller, T.F. A critical review of thermal issues in lithium-ion batteries. *J. Electrochem. Soc.* **2011**, *158*, R1–R25. [CrossRef]
46. Rao, Z.; Wang, S. A review of power battery thermal energy management. *Renew. Sustain. Energy Rev.* **2011**, *15*, 4554–4571. [CrossRef]
47. Xia, G.; Cao, L.; Bi, G. A review on battery thermal management in electric vehicle application. *J. Power Sources* **2017**, *367*, 90–105. [CrossRef]
48. Li, W.; Rentemeister, M.; Badeda, J.; Jost, D.; Schulte, D.; Sauer, D.U. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *J. Energy Storage* **2020**, *30*, 101557. [CrossRef]

49. Wang, Y.; Xu, R.; Zhou, C.; Kang, X.; Chen, Z. Digital twin and cloud-side-end collaboration for intelligent battery management system. *J. Manuf. Syst.* **2022**, *62*, 124–134. [CrossRef]
50. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6730–6739. [CrossRef]
51. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *J. Power Sources* **2018**, *400*, 242–255. [CrossRef]
52. Bian, C.; He, H.; Yang, S.; Huang, T. State-of-charge sequence estimation of lithium-ion battery based on bidirectional long short-term memory encoder-decoder architecture. *J. Power Sources* **2020**, *449*, 227558. [CrossRef]
53. Bian, C.; He, H.; Yang, S. Stacked bidirectional long short-term memory networks for state-of-charge estimation of lithium-ion batteries. *Energy* **2020**, *191*, 116538. [CrossRef]
54. Kollmeyer, P.J.; Vidal, C.; Naguib, M.; Skells, M. LG 18650HG2 Li-ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script. *Mendeley Data* **2020**. Available online: https://data.mendeley.com/datasets/cp3473x7xv/3 (accessed on 2 October 2026).
55. Tian, Y.; Lai, R.; Li, X.; Xiang, L.; Tian, J. A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter. *Appl. Energy* **2020**, *265*, 114789. [CrossRef]
56. Yang, F.; Zhang, S.; Li, W.; Miao, Q. State-of-charge estimation of lithium-ion batteries using LSTM and UKF. *Energy* **2020**, *201*, 117664. [CrossRef]
57. Liu, Y.; Shu, X.; Yu, H.; Shen, J.; Zhang, Y.; Liu, Y.; Chen, Z. State of charge prediction framework for lithium-ion batteries incorporating long short-term memory network and transfer learning. *J. Energy Storage* **2021**, *37*, 102494. [CrossRef]
58. Tian, Y.; Lai, R.; Li, X.; Tian, J. State-of-charge estimation for lithium-ion batteries based on attentional sequence-to-sequence architecture. *J. Energy Storage* **2023**, *62*, 106836. [CrossRef]
59. Tian, J.; Chen, C.; Shen, W.; Sun, F.; Xiong, R. Deep Learning Framework for Lithium-ion Battery State of Charge Estimation: Recent Advances and Future Perspectives. *Energy Storage Mater.* **2023**, *61*, 102883. [CrossRef]
60. Rastegarpanah, A.; Hathaway, J.; Ahmeid, M.; Lambert, S.; Walton, A.; Stolkin, R. A rapid neural network–based state of health estimation scheme for screening of end of life electric vehicle batteries. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 330–346. [CrossRef]
61. Xiong, R.; Li, L.; Tian, J. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. *J. Power Sources* **2018**, *405*, 18–29. [CrossRef]
62. Richardson, R.R.; Osborne, M.A.; Howey, D.A. Gaussian process regression for forecasting battery state of health. *J. Power Sources* **2017**, *357*, 209–219. [CrossRef]
63. Yang, D.; Zhang, X.; Pan, R.; Wang, Y.; Chen, Z. A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *J. Power Sources* **2018**, *384*, 387–395. [CrossRef]
64. Severson, K.A.; Attia, P.M.; Jin, N.; Perkins, N.; Jiang, B.; Yang, Z.; Chen, M.H.; Aykol, M.; Herring, P.K.; Fraggedakis, D.; et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **2019**, *4*, 383–391. [CrossRef]
65. Zhang, Y.; Tang, Q.; Zhang, Y.; Wang, J.; Stimming, U.; Lee, A.A. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. *Nat. Commun.* **2020**, *11*, 1706. [CrossRef] [PubMed]
66. Lu, J.; Xiong, R.; Tian, J.; Wang, C.; Sun, F. Deep learning to estimate lithium-ion battery state of health without additional degradation experiments. *Nat. Commun.* **2023**, *14*, 2760. [CrossRef] [PubMed]
67. Ma, G.; Xu, S.; Jiang, B.; Cheng, C.; Yang, X.; Shen, Y.; Yang, T.; Huang, Y.; Ding, H.; Yuan, Y. Real-time personalized health status prediction of lithium-ion batteries using deep transfer learning. *Energy Environ. Sci.* **2022**, *15*, 4083–4094. [CrossRef]
68. Pang, H.; Qu, X.; Chen, K.; Yan, X.; Burke, A.F.; Zhao, J. A transferable deep learning framework for cross-chemistry and cross-scale state of health estimation in lithium-ion batteries. *Energy* **2026**, *354*, 141079. [CrossRef]
69. Jing, H.; Hu, J.; Ou, S.; Lv, Z.; Lyu, R.; Zhao, J. Scalable and generalizable deep learning for battery state of health estimation in on-road electric vehicles. *J. Energy Chem.* **2025**, *110*, 823–841. [CrossRef]
70. Chen, Y.; Wu, Y.; Li, C. A cycle-aware and physics-informed framework for battery remaining useful life prediction. *Sci. Rep.* **2025**, *15*, 45775. [CrossRef] [PubMed]
71. Zhao, Z.; Feng, M.; Yu, Q.; Yuan, H.; Wang, F.; Zhai, Z. Towards trustworthy lithium-ion battery remaining useful life prediction with uncertainty quantification. *J. Energy Storage* **2026**, *167*, 122408. [CrossRef]
72. Mao, R.; Han, P.; Xu, W.; Yuan, N.; Zhang, H. Lithium-ion battery remaining useful life prediction using Bayesian filters with neural network pseudo observations. *J. Energy Storage* **2026**, *146*, 120179. [CrossRef]
73. Dong, X.; Yao, M.; Xiang, S. A load-balanced deformable expert network for lithium-ion battery remaining useful life prediction. *J. Energy Storage* **2026**, *141*, 119587. [CrossRef]
74. Xiao, Y.; Zhu, X.; Wu, J.; Luo, J.; Li, Q.; Xiong, R.; Chen, W. A multi-stage augmentative generalization learning prediction model for lithium-ion battery remaining useful life under uncertain working conditions. *Energy* **2025**, *335*, 138228. [CrossRef]

75. Rizanov, S.; Stoynova, A.; Mihov, G. Toward trustworthy and transferable SOH/RUL estimation for lithium-ion batteries: A critical review and multi-fidelity validation framework from laboratory cells to real-world packs. *Batteries* **2026**, *12*, 255. [CrossRef]
76. Naresh, G.; Praveenkumar, T.; Madheswaran, D.K.; Varuvel, E.G.; Pugazhendhi, A.; Thangamuthu, M.; Muthiya, S.J. Battery fault diagnosis methods for electric vehicle lithium-ion batteries: Correlating codes and battery management system. *Process Saf. Environ. Prot.* **2025**, *196*, 106919. [CrossRef]
77. Li, H.; Kaleem, M.B.; Liu, K.; Wu, Y.; Liu, W.; Peng, Q. Fault prognosis of Li-ion batteries in electric vehicles: Recent progress, challenges and prospects. *J. Energy Storage* **2025**, *116*, 116002. [CrossRef]
78. Zhao, J.; Feng, X.; Tran, M.K.; Fowler, M.; Ouyang, M.; Burke, A.F. Battery safety: Fault diagnosis from laboratory to real world. *J. Power Sources* **2024**, *598*, 234111. [CrossRef]
79. Kaleem, M.B.; Zhou, Y.; Jiang, F.; Liu, Z.; Li, H. Fault detection for Li-ion batteries of electric vehicles with segmented regression method. *Sci. Rep.* **2024**, *14*, 31922. [CrossRef] [PubMed]
80. Naguib, M.; Chen, J.; Kollmeyer, P.; Emadi, A. Thermal fault detection of lithium-ion battery packs through an integrated physics and deep neural network based model. *Commun. Eng.* **2025**, *4*, 79. [CrossRef] [PubMed]
81. Fan, Y.; Huang, Z.; Li, H.; Yuan, W.; Yan, L.; Liu, Y.; Chen, Z. Fault detection for Li-ion batteries of electric vehicles with feature-augmented attentional autoencoder. *Sci. Rep.* **2025**, *15*, 18534. [CrossRef] [PubMed]
82. Yu, H.; Zhang, H.; Zhang, Z.; Yang, S. State estimation of lithium-ion batteries via physics-machine learning combined methods: A methodological review and future perspectives. *eTransportation* **2025**, *24*, 100420. [CrossRef]
83. Yang, H.; Tian, J.; Mai, W.; Wang, C.; Ran, L.; Wu, T.; He, S.; Wang, Z.; Shi, X.; Liang, Z.; et al. Privacy-preserving collaborative battery fault warning for massive electric vehicles by heterogeneous data from charging stations. *Nat. Commun.* **2026**, *17*, 974. [CrossRef] [PubMed]
84. Sheng, L.; Lan, H.; Tao, B.; Zhang, Z.; Wang, H. Anisotropic thermal conductivity quantification in prismatic lithium-ion batteries. *J. Energy Storage* **2026**, *162*, 122112. [CrossRef]
85. Lei, S.; Huaiyu, L.; Bojun, T.; Chunfeng, Z.; Xiaojun, Z.; Liyang, W. Experiments on liquid-immersed thermal runaway management for large energy-stored lithium-ion battery modules. *Therm. Sci. Eng. Prog.* **2025**, *68*, 104347. [CrossRef]
86. Qi, W.; Yang, J.; Zhang, Z.; Wu, J.; Lan, P.; Xiang, S. Investigation on thermal management of cylindrical lithium-ion batteries based on interwound cooling belt structure. *Energy Convers. Manag.* **2025**, *340*, 119962. [CrossRef]
87. Xia, X.; Huang, Y.; Gao, J.; Zhang, Y. Reinforcement learning for integrated thermal management of electric vehicle batteries and air conditioning systems. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2026**, *240*, 6056–6071. [CrossRef]
88. Vieira, R.; Kollmeyer, P.; Pitault, L.; Emadi, A. Comprehensive comparison of machine learning and Kalman filter battery state of charge estimators. *IEEE Access* **2025**, *13*, 36321–36338. [CrossRef]
89. Sohail, M.; Tanveer, M.; Kim, H.S. Trustworthy machine learning and mathematical modelling for lithium-ion battery state-of-health estimation. *Mathematics* **2026**, *14*, 1879. [CrossRef]
90. Galiounas, E.; Owen, R.E.; Robinson, J.B.; Jervis, R. The generalisation challenge: Assessment of the efficacy of acoustic signals for state estimation of lithium-ion batteries via machine learning. *J. Power Sources* **2025**, *630*, 236047. [CrossRef]
91. Abadade, Y.; Temouden, A.; Bamoumen, H.; Benamar, N.; Chtouki, Y.; Hafid, A.S. A comprehensive survey on TinyML. *IEEE Access* **2023**, *11*, 96892–96922. [CrossRef]
92. Weng, Y.; Guan, W.; Ababei, C. CNN-based prediction of multi-variables for lithium-ion batteries optimized with TinyML and deployed on edge devices. *J. Energy Storage* **2026**, *141*, 119508. [CrossRef]
93. Keshyagol, K.; Kulkarni, S. Artificial intelligence embedded platform for battery management and autonomous navigation in a prototype electric vehicle. *Discov. Sustain.* **2025**, *6*, 1474. [CrossRef]
94. Kim, J.M. Explainable Artificial Intelligence (XAI) in battery management systems: A comprehensive review. *Int. J. Sci. Technol. Res. Arch.* **2025**, *8*, 14–26. [CrossRef]
95. Divya, G.; Mustare, N.; Kumar, K.P. Improving electric vehicle battery health management with explainable AI. *E3S Web Conf.* **2025**, *616*, 03030. [CrossRef]
96. *ISO 26262:2018*; Road Vehicles—Functional Safety. ISO: Geneva, Switzerland, 2018.
97. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
98. *Regulation (EU) 2024/1689*; Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). European Parliament; Council of the European Union: Bruxelles, Belgium, 2024. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689 (accessed on 1 October 2026).
99. Hu, X.; Jiang, X.-N.; Zhang, J.; Wang, S.; Zhou, M.; Zhang, B.; Gan, Z.; Yu, B.-X. Electric vehicle charging network security: A survey. *J. Syst. Archit.* **2025**, *159*, 103337. [CrossRef]
100. Dehrouyeh, F.; Shaer, I.; Nikan, S.; Ajaei, F.; Shami, A. TinyML-enabled resource-efficient framework for real-time network securiy in electric vehicle charging networks. *IEEE Trans. Netw. Sci. Eng.* **2026**, *13*, 5092–5109. [CrossRef]
101. *ISO/SAE 21434:2021*; Road Vehicles—Cybersecurity Engineering. ISO: Geneva, Switzerland, 2021.

102. *UN Regulation No. 155*; Cyber Security and Cyber Security Management System. United Nations: Geneva, Switzerland, 2021.
103. *UN Regulation No. 156*; Software Update and Software Update Management System. United Nations: Geneva, Switzerland, 2021.
104. Tesla, Inc. *Impact Report 2024*; Tesla, Inc.: Austin, TX, USA, 2025. Available online: https://www.tesla.com/ns_videos/2024-tesla-impact-report.pdf (accessed on 2 October 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.