Subsystem-oriented condition monitoring for connected vehicles using unlabelled data

Book of Proceedings

of the 7th edition of the International Conference on Quality Engineering and Management

ICQEM 2026

7th International Conference on Quality Engineering and Management

2-3 July 2026, Universidade NOVA de Lisboa, Lisbon, Portugal

Subsystem-oriented condition monitoring for connected vehicles using unlabelled data

Stefano Puttero [1], Elisa Verna [1], Gianfranco Genta [1] and Maurizio Galetto [1]

[1] Politecnico di Torino, Department of Management and Production Engineering, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; stefano.puttero@polito.it

Abstract — Connected vehicles continuously generate large volumes of telemetry data, but it remains difficult to transform these streams into useful maintenance support when failure labels are scarce, degradation is gradual and heterogeneous, and operators require stable, interpretable and operationally manageable outputs. This paper presents a subsystem-oriented monitoring architecture for electric vehicles that is designed for this setting. Instead of collapsing the vehicle into a single global health score, the proposed approach monitors four key subsystems (battery, drivetrain, brakes and tyres/suspension) and computes for each of them a Subsystem Condition Indicator (SCI) referenced to subsystem-specific baseline behaviour. The SCI combines structural deviation, derived from reconstruction error, with short-term temporal deviation, derived from next-step forecasting error. It is then converted into stable alert states through fixed, baseline-derived thresholds, persistence logic and hysteresis. Thus, the architecture transforms unlabelled telemetry data into dashboard-oriented outputs that facilitate day-to-day monitoring, including subsystem condition, short-term trends, alarm states and concise reason-code-based cues to support initial inspections. A case study based on an electric-vehicle telemetry dataset shows that the proposed architecture preserves subsystem differentiation, produces interpretable and manageable alert behaviour, and remains stable under alternative baseline definitions and moderate missing-data perturbations, while also providing a basis for future extensions toward more prescriptive maintenance support.

Keywords – Maintenance, Condition monitoring, Fleet management, Connected vehicles

I. Introduction

The growing number of connected vehicles is expanding the scope of data-driven maintenance and Prognostics and Health Management (PHM) strategies based on operational data. Instead of being triggered mainly by inspections, complaints or failures, maintenance can now be supported by continuous telemetry streams generated during vehicle operation (Hu *et al.*, 2022). Modern Electric Vehicles (EVs) collect electrical, thermal, mechanical and contextual data over long operating periods, creating favourable conditions for data-driven condition monitoring. In practice, however, many organisations still struggle to convert this telemetry into maintenance decisions that are timely, trustworthy and economically meaningful (Mahale *et al.*, 2025). The problem is often not the lack of data itself, but the lack of monitoring outputs that align with the way maintenance decisions are made (Theissler *et al.*, 2021).

This challenge is particularly evident in fleet-based business models. Shared mobility operators, leasing companies, corporate fleets and service providers are directly exposed to the operational and economic consequences of unexpected vehicle degradation, downtime and unplanned maintenance (Sharma, 2024). They therefore require tools that can identify which vehicles require attention first, which subsystems are behaving abnormally and whether the condition observed is stable enough to justify inspection or intervention (Benhanifia *et al.*, 2025). An exclusively diagnostic approach based on isolated threshold violations can result in numerous unrelated alerts, whereas a purely prognostic approach based on time-to-failure or Remaining Useful Life (RUL) can be challenging to implement when the historical data required for supervised modelling is limited, inconsistent or unavailable (Zonta *et al.*, 2020). Reviews of predictive maintenance in the automotive sector explicitly identify data availability and quality as key challenges. Recent fleet-oriented research has also highlighted the need for unsupervised monitoring due to the frequent absence of accurate labels (Theissler *et al.*, 2021).

Electric vehicles highlight the tension between data richness and label scarcity. The main subsystems of these vehicles evolve under operating conditions that are strongly context-dependent, and their degradation patterns are often non-linear and asynchronous (Severson *et al.*, 2019). Battery-related variables, drivetrain signals, braking behaviour and tyre-related measurements are all influenced by usage conditions, load and the environment. Consequently, the same absolute sensor value can have different meanings in different contexts, limiting the usefulness of fixed thresholds and motivating monitoring strategies based on deviations from learned reference behaviour. The need for broader, context-aware, data-driven monitoring is consistent with recent work on AI-enabled EV management and unsupervised fleet maintenance approaches that explicitly model operational context (Swaroopa Rani and Jatoth, 2025).

Another practical issue concerns the level of aggregation. Vehicle-level condition scores are appealing because they provide a concise overview and support high-level reporting. However, for maintenance engineers and fleet coordinators, a single score can obscure the most relevant information for taking action (Theissler *et al.*, 2021). When a vehicle starts to deteriorate, the first operational question is rarely whether the vehicle as a whole is healthy. More useful questions are which subsystem deserves attention, how quickly its condition is changing, and whether the signal is persistent enough to justify action. For this reason, a monitoring architecture that aggregates data too much may reduce the practical value of the available telemetry. While existing work on fleet-level unsupervised predictive maintenance demonstrates the importance of asset-level ranking and anomaly detection (Mahale *et al.*, 2025), there is still scope for solutions that provide clearer, more subsystem-oriented support for maintenance triage.

This paper addresses these issues by redefining the problem as the generation of stable and interpretable subsystem signals from unlabelled telemetry. The objective is not to introduce a radically new anomaly-detection algorithm or to perform traditional supervised prognostics. Instead, it proposes integrating standard learning-based modelling blocks, trained on baseline data representative of nominal behaviour, into a hybrid monitoring architecture that combines learned anomaly representations, statistical deviation analysis, and operational alarm governance logic. This architecture could form the analytical core of a dashboard for connected vehicle maintenance and support subsequent operational monitoring. In PHM terms, this work is more closely aligned with operational health monitoring and maintenance decision support than with label-intensive failure prediction (Hu *et al.*, 2022). For this reason, the proposed approach is built around four practical requirements: subsystem specificity, baseline-referenced interpretation, stable alarm governance and concise, drill-down information for inspection prioritisation. Each major subsystem is monitored via a dedicated Subsystem Condition Indicator (SCI), which detects structural deviations in multivariate sensor states and temporal deviations in their short-term evolution. These signals are then smoothed, thresholded, and translated into alarm states that are suitable for operational interpretation.

The remainder of the paper is organised as follows. Section II explains the research gap and clarifies why a subsystem-oriented operational approach is necessary. Section III presents the monitoring architecture, emphasising the design choices that enable deployment. Section IV describes the case study and the implementation logic used to test the architecture with electric-vehicle telemetry data. Section V discusses the main results and their implications for maintenance and fleet management. Section VI concludes the paper and outlines the next steps for multi-vehicle validation.

II. Research gap and motivation

Recent literature on predictive maintenance in the automotive sector reveals a growing interest in data-driven diagnostics, fault detection, and RUL estimation (Mahale *et al.*, 2025). However, much of this literature still relies on supervised learning settings where labelled fault events, degradation stages or run-to-failure histories are assumed to be available (Serradilla *et al.*, 2022). In practice, this assumption is often difficult to fulfil in connected-vehicle applications, where maintenance records are frequently incomplete, inconsistent, or misaligned with the actual progression of subsystem degradation (Hu *et al.*, 2022). A survey focusing on machine learning–enabled predictive maintenance in the automotive sector explicitly discusses data availability and quality as key challenges for deployment. More recent works on RUL estimation are increasingly motivated by the need for self-supervised or label-efficient solutions, precisely because labelled degradation data are limited (Theissler *et al.*, 2021).

This limitation is particularly relevant in vehicle settings with a high volume of telemetry data. Connected vehicles generate large volumes of multivariate time-series data, but these streams are not automatically accompanied by reliable labels that can support supervised prognostics (Swaroopa Rani and Jatoth, 2025). Even when maintenance records exist, they do not necessarily indicate the precise moment of failure onset, nor do they always depict the entire process from normal behaviour to degradation and failure. Consequently, establishing a robust supervised pipeline for fault classification or RUL prediction can be impractical, particularly when the available data covers only one or a few vehicles or when there are no explicit fault annotations (Lipu *et al.*, 2018). In such cases, methods that learn nominal behaviour directly from unlabelled operational data are a more practical option. This approach is consistent with recent work on unsupervised predictive maintenance for fleet management, where anomaly detection is used to address the issue of scarce labels and highly variable operating contexts (Giannoulidis and Gounaris, 2023).

At the same time, shifting from supervised prognostics to unsupervised monitoring does not automatically solve the deployment problem. While existing unsupervised monitoring and anomaly-detection approaches are useful when labelled failures are scarce, they often remain centred on deviation detection, generic anomaly scores, or asset-level ranking (Serradilla *et al.*, 2022). In practice, such outputs can be challenging to act upon as they do not always indicate which subsystem requires attention, whether the deviation is significant enough to warrant inspection and how alerts should be managed. Therefore, maintenance personnel need more than just knowledge of unusual behaviour; they require outputs with stable semantics, understandable scales, and clear alert conditions. Recent work on explainable anomaly detection for predictive maintenance explicitly acknowledges that limited interpretability and reduced user trust are significant barriers to adoption. This suggests that the practical value of an unsupervised monitoring method depends on two things: its ability to detect deviations and its ability to translate these deviations into condition signals that can be consistently interpreted over time (Resanovic *et al.*, 2026).

A second gap concerns the level at which monitoring results are presented (Benhanifia *et al.*, 2025). While many existing approaches provide asset-level anomaly outputs, which can be useful for generic ranking or screening, they are often insufficient for making maintenance decisions in vehicles. In practice, inspections and interventions are usually organised around subsystems and components, such as the battery, drivetrain, braking system, or tyre and suspension systems. While a single vehicle-level score may indicate that something abnormal is happening, it does not necessarily provide the required level of detail to support triage, prioritisation and follow-up inspection (Mahale *et al.*, 2025). While the literature on fleet-oriented unsupervised maintenance recognises the importance of context-aware and streaming monitoring, there is still scope for solutions that connect anomaly detection more directly to operational actions at the subsystem level (Giannoulidis and Gounaris, 2023).

A third gap relates to the robustness of real telemetry streams. Reports from predictive maintenance reviews consistently indicate that real-world data is affected by noise, missing values, variability across operating conditions and other quality issues that complicate operational deployment (Taylor *et al.*, 2016). This is particularly pertinent in the context of connected vehicles, where measurements are influenced by usage patterns, ambient conditions and the occasional loss of data. In this context, a monitoring architecture must be able to detect deviations, remain stable enough to avoid oscillating alerts, and be interpretable enough to preserve operator trust. Therefore, calibration, alert persistence, and alarm-clearing logic are integral to the monitoring process itself (Nunes *et al.*, 2023).

In light of these considerations, there is still a gap between unsupervised telemetry analytics and monitoring solutions that can be effectively used in the maintenance of connected vehicles. While recent studies have demonstrated the value of label-free and context-aware monitoring (Lipu *et al.*, 2018), less attention has been given to structuring such outputs for stable operational interpretation. This is particularly important when maintenance decisions require subsystem-level visibility rather than generic vehicle-level anomaly evidence. This is especially critical in situations where reliable labels are unavailable, and monitoring outputs must support triage, inspection planning, and intervention prioritisation. Against this background, this paper proposes a monitoring architecture for subsystems based on unlabelled telemetry. Instead of introducing a new anomaly-detection algorithm, it integrates established unsupervised learning components with baseline-referenced statistical monitoring and alarm governance logic. In this architecture, deviation signals are consolidated into the SCI indicator, which is designed for operational use. The work thus aims to translate anomaly evidence into a more structured and actionable representation to support maintenance.

III. Subsystem condition indicators

The proposed monitoring architecture is organised into four subsystem groups reflecting both engineering structure and maintenance relevance: battery, drivetrain, brakes, and tyres/suspension. This grouping is intentionally simple and transparent, ensuring that the analytical outputs remain aligned with typical maintenance planning and execution (Zonta *et al.*, 2020). Instead of monitoring the vehicle as a single, aggregated asset, the architecture preserves visibility at the subsystem level, which is more consistent with inspection planning, prioritisation, and maintenance triage. In addition to signals specific to each subsystem, the monitoring logic incorporates contextual variables such as ambient temperature, humidity, speed, load, idle time, distance travelled, and route roughness. Accordingly, the objective is not to flag demanding but legitimate usage, but to identify behaviour that is unusual relative to the operating context.

Fig. 1 summarises the overall workflow of the proposed subsystem-oriented monitoring architecture. Starting with EV telemetry data, the pipeline first groups the available variables by subsystem, before extracting two complementary forms of deviation evidence: a static component obtained from autoencoder (AE) reconstruction error and a dynamic component obtained from Gated Recurrent Unit (GRU)-based next-step forecasting error. These two signals are then fused and normalised with respect to subsystem-specific baseline behaviour to obtain a bounded SCI. The SCI is then translated into stable alert states via the alert governance logic and supplemented with inspection-support outputs.

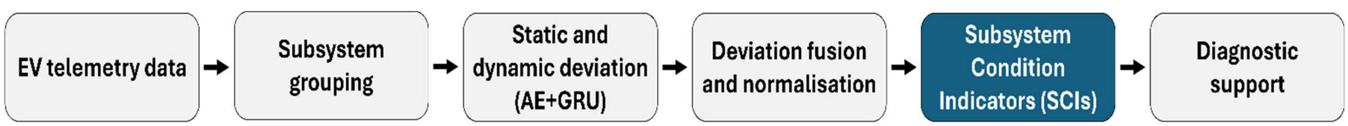

Fig. 1. Subsystem-oriented operational monitoring workflow.

To clarify the learning-based nature of the proposed approach, the monitoring architecture has been organised into two distinct stages: an offline model training stage, and an online operational monitoring stage. During the initial stage, the AE and GRU models are trained using baseline data that is representative of the nominal behaviour of the subsystem. The AE learns compact representations of nominal multivariate patterns and produces reconstruction errors when incoming observations deviate from this learned behaviour. Similarly, the GRU model learns the temporal evolution of nominal signals and provides forecasting errors when observed dynamics depart from expected temporal patterns. Once trained, these models are deployed for operational monitoring in the second stage, where reconstruction and forecasting errors are computed on new telemetry streams and converted into condition indicators and alerts oriented towards the subsystem.

Therefore, the framework should be interpreted as a hybrid monitoring architecture rather than a purely learning-driven anomaly detection model. Learning-based components provide compact representations of nominal multivariate and temporal behaviour. Meanwhile, statistical monitoring and alarm governance components transform the resulting deviation measures into stable, interpretable, subsystem-level condition indicators. Table I summarises the functional decomposition, distinguishing the learning-based components from the statistical monitoring and operational alarm governance logic.

TABLE I. Functional role of the main components in the proposed hybrid monitoring architecture.

Component	*Role in the framework*	*Type of logic*
AE model	Learns nominal multivariate relationships and produces reconstruction-error deviations	Learning-based
GRU model	Learns short-term temporal dynamics and produces forecasting-error deviations	Learning-based
Error calibration and robust scaling	Convert model errors into comparable subsystem-level deviation scores	Statistical monitoring
Smoothing and thresholds	Reduce short-term fluctuations and support stable interpretation of deviations	Statistical monitoring

Component	Role in the framework	Type of logic
Alarm governance	Translates deviation patterns into operational alerts and inspection-oriented information	Heuristic/ operational logic

In the case study, the same modelling logic was applied to each subsystem independently. Prior to training, the selected subsystem variables were robustly scaled using baseline data so that deviations could be compared across heterogeneous signals. The AE was used to reconstruct the current state of the subsystem, while the GRU model was used to forecast its short-term evolution. The resulting reconstruction and forecasting errors were calibrated with respect to the baseline distribution and converted into SCI. The main implementation settings of the learning-based components are reported in Table II. The training used an 80%-20% partition of the baseline data in chronological order, with early stopping based on the validation loss. This configuration was used for all the subsystem-specific models.

TABLE II. MAIN IMPLEMENTATION SETTINGS OF THE LEARNING-BASED COMPONENTS.

Aspect	AE	GRU
Input	Current subsystem feature vector	Rolling subsystem feature sequence
Main role	Reconstruction of nominal subsystem states	Short-term forecasting of subsystem dynamics
Training data	Baseline nominal data	Baseline nominal data
Architecture	Dense encoder-decoder with 64–32–16 bottleneck	GRU layer with 64 hidden units and dense output
Error used	Mean absolute reconstruction error	Mean absolute one-step forecasting error
Use in SCI	Structural deviation component	Temporal deviation component

The second step in the workflow is to define subsystem-specific feature groups. These groups reflect the physical structure of the vehicle and the operational logic of maintenance activities. Battery-related signals capture electrochemical and thermal behaviour, drivetrain variables describe mechanical and energetic performance, brake-related features reflect braking workload and efficiency, and the tyres/suspension group captures conditions associated with load transfer, handling, and chassis behaviour. While the exact features available depend on the telemetry dataset being analysed, Table III summarises the representative variable groups considered by the architecture for each subsystem (Taylor *et al.*, 2016).

TABLE III. REPRESENTATIVE VARIABLE GROUPS CONSIDERED BY THE MONITORING ARCHITECTURE.

Subsystem	Representative variables	Operational relevance
Battery	State of Charge (SoC), State of Health (SoH), battery voltage/current, battery temperature, charge cycles, contextual variables	Supports monitoring of electrochemical stress, thermal behaviour, and charging-related degradation
Drivetrain	Motor temperature, vibration, torque, revolutions per minute (RPM), power consumption, contextual variables	Captures efficiency losses, abnormal load transfer, and evolving dynamic behaviour
Brakes	Brake pad wear, pressure, brake temperature, regenerative braking efficiency, contextual variables	Links alerts to braking workload and possible imbalance between regenerative and friction braking
Tyres/Suspension	Tyre pressure, tyre temperature, suspension load, contextual variables	Provides a route to identify handling, load-distribution, and chassis-related issues

The monitoring workflow then proceeds to identify a baseline specific to the subsystem in question, i.e. an initial reference window representing its normal behaviour. In realistic deployment scenarios, nominal periods are rarely labelled explicitly, meaning that it is not possible to select the reference behaviour manually in a reliable or scalable way (Jardine *et al.*, 2006). Therefore, the pipeline evaluates several candidate baseline windows located at the beginning of the time series. These include fixed fractions of the initial history and a scan window representing the initial operating period. For each candidate, a robust sanity score is computed based on the proportion of extreme values observed across the relevant variables, using interquartile-range fences. The candidate with the lowest score is then selected as the baseline for that subsystem. This reduces analyst subjectivity and recognises that different subsystems may stabilise at different rates, even within the same vehicle.

Once the baseline specific to the subsystem has been identified, the data preparation stage is designed to preserve temporal consistency and prevent information leakage from later, potentially degraded operating periods into the nominal reference. First, numeric variables are converted into a consistent numerical format. Then, missing values are managed using a leakage-safe procedure: forward filling is employed to preserve temporal continuity and any remaining missing values at the beginning of the acquisition history are replaced with medians estimated from only the selected baseline rows. This strategy is intentionally conservative. The aim is not to reconstruct an ideal signal, but to ensure that the reference remains anchored to nominal behaviour. After imputation, the variables are scaled using robust statistics derived from the baseline (i.e. medians and interquartile ranges) so that heterogeneous sensor units can be compared while limiting sensitivity to outliers (Chandola *et al.*, 2009). The main choices regarding preprocessing and calibration were made to prioritise operational robustness rather than optimising performance against labelled fault outcomes, which are unavailable in this setting. Robust scaling, baseline-derived calibration, quantile-based thresholds and persistence-based alarm activation were therefore adopted to reduce sensitivity to outliers, limit unstable alerts and preserve interpretability across subsystems. The same parameter-selection logic was consistently applied to all subsystem-specific models.

After the preparation of the data, the SCI is constructed by combining two complementary deviation channels. The first of these is a structural deviation channel, which captures how far the current multivariate sensor configuration deviates from patterns observed during nominal operation. In the current implementation, this component is obtained from an AE that has been trained using only baseline data (Heger *et al.*, 2020). Under reference conditions, the reconstruction error is expected to remain relatively small. However, when the subsystem moves away from the learned nominal patterns, the reconstruction error increases. This first channel is useful for identifying atypical states, such as unusual combinations of electrical or mechanical variables that may be associated with emerging degradation.

This structural view is complemented by a temporal deviation channel. Although reconstruction error indicates whether the current state is unusual, it does not explicitly capture whether the short-term evolution of the subsystem is becoming abnormal. For this reason, the architecture also includes a next-step predictor that is trained using baseline sequences. In the reference implementation, this predictor is a GRU model which estimates the expected change in the scaled sensor vector over the next time step using a rolling lookback window (Chung *et al.*, 2014). The resulting prediction error quantifies how unusual the local dynamics are relative to normal behaviour. This temporal component is particularly important when the first signs of degradation manifest as a change in short-term evolution and not as an obvious, instantaneous abnormality.

As a monitoring pipeline intended for practical use should remain functional under non-ideal conditions, the implementation includes controlled fallback options. For example, if the AE cannot be reliably trained or the available baseline is too short, the structural deviation channel can revert to a Principal Component Analysis (PCA)-based reconstruction model. Similarly, if the recurrent predictor cannot be trained under the available conditions, the temporal channel can revert to a ridge regression predictor. These alternatives are not intended to replace the preferred models in all situations, but to ensure continuous monitoring and prevent unreliable behaviour when the pipeline is first used, when there is limited data, or when computer settings are restricted.

Once the two deviation channels have been normalised and fused, the resulting signal is mapped onto the SCI via a calibrated sigmoid transformation (Warton, 2008):

$$\mathrm{SCI}(t) = \frac{1}{1+e^{\beta\left(z(t)-z_{ref}\right)}} \tag{1}$$

where $z(t)$ is the fused deviation signal at time t and z_{ref} is the baseline reference level, β is a calibration parameter estimated from baseline behaviour. This mapping is important because it enables maintenance users to interpret a bounded and monotonic condition scale more easily than raw model errors. In the implemented formulation, nominal operation remains in a clearly non-critical state while increasing deviation pushes the SCI towards lower values. Therefore, the SCI is an indicator of relative condition, based on learned reference behaviour, and not as a direct measure of probability of failure.

As telemetry streams may contain local fluctuations and short-lived excursions (Mahale *et al.*, 2025), the SCI is smoothed using an exponentially weighted moving average before alert generation. This improves readability, stabilises trend interpretation and enables more reliable alarm governance. The smoothed SCI is then translated into discrete operating states using thresholds derived from the SCI distribution observed during the baseline period. In practice, the baseline is used to estimate the range of values associated with normal subsystem behaviour, and fixed, quantile-based cut-offs are extracted from the resulting distribution. These cut-offs define a warning state and a more severe critical state. The warning state highlights behaviour that requires attention, even if it is not severe enough to justify an alarm. The critical threshold, on the other hand, is used for alarm activation. To avoid alarms caused by isolated fluctuations, activation is persistence-based: the SCI must remain below the critical threshold for a predefined number of consecutive time steps before the alarm is raised. The process of alarm clearance is governed by hysteresis, which signifies that recovery necessitates the SCI to rise above a less severe clearance threshold, as opposed to merely crossing back over the alarm-on threshold. This threshold design reduces chattering and limits alarm fatigue, making the alerting logic more suitable for day-to-day maintenance support.

At each telemetry step, the tool provides a concise set of maintenance-oriented monitoring outputs for each subsystem. These include the current SCI value, the current alarm state, the short-term SCI trend, and the threshold band used for interpretation. The architecture also provides simple reason codes, obtained by aggregating absolute reconstruction residuals during alarm periods and identifying the variables with the highest average contribution. These reason codes should be interpreted as inspection-support cues rather than root-cause identifications. They are intended to improve operational clarity and facilitate maintenance triage by indicating which variables and subsystem groups may require inspection first. In this sense, the proposed monitoring layer can also be seen as a first step towards supporting prescriptive maintenance (Ansari *et al.*, 2019), since persistent alarms and their associated inspection-support cues may later be linked to recommended inspection or intervention actions. The outputs returned by the pipeline in dashboard format at each time step are summarised in Table IV.

TABLE IV. OPERATIONAL OUTPUTS PROVIDED BY THE SUBSYSTEM MONITORING PIPELINE.

Pipeline output	***Generated by the pipeline***	***Interpretation***	***Maintenance use***
SCI value	Bounded SCI in [0,1]	Lower values indicate stronger deviation from baseline behaviour	Quick view of current subsystem status
Trend over time	Gradient of the SCI stream	Negative trend indicates worsening behaviour	Prioritise assets before the condition becomes persistent
Alarm state	Persistence and hysteresis applied to the SCI	Binary indication of active/non-active critical condition	Stable trigger for inspection or escalation

Pipeline output	Generated by the pipeline	Interpretation	Maintenance use
Reason codes	Top residual contributors during alarm windows	Inspection-support cues associated with subsystem deviations	Guide first inspection checks and maintenance triage

IV. CASE STUDY

The proposed monitoring architecture was evaluated using the publicly available EVIoT Predictive Maintenance dataset on Kaggle (Kaggle, 2025). This electric-vehicle telemetry dataset is sampled at 15-minute intervals and includes electrical, thermal, mechanical and operational measurements. It represents a realistic connected-vehicle monitoring scenario in which explicit fault labels are unavailable for supervised modelling. Consequently, the case study does not seek to quantify classical fault-detection accuracy, the false-positive rate or time to detection. Instead, the evaluation focuses on the operational behaviour of the monitoring architecture. This includes the stability of SCI trajectories, manageable alarm rates, sensitivity to baseline definition and robustness under moderate missing-data perturbations. These criteria were selected because they are directly related to deployment usability in label-scarce monitoring settings. Thus, the investigation's objective is to evaluate if the suggested subsystem-level framework is capable of producing significant, practically beneficial maintenance assistance results from unlabelled telemetry.

A first indication of applicability is provided by the subsystem SCI trajectories in Fig. 2, where the battery, drivetrain, brakes, and tyres/suspension subsystems are tracked together with the alarm-state markers and the fixed alarm-on and clearance thresholds. In this case study, the two thresholds are derived from the baseline SCI distribution using the 0.25 and 0.35 quantiles, respectively, to define conservative alarm and recovery regions. This representation preserves subsystem-specific behaviour, rather than collapsing the monitored asset into a single aggregate signal. This makes it possible to distinguish localised deviations from more widespread abnormal behaviour within the same operating period. At the same time, the plotted alarm states reflect the persistence-and-hysteresis logic adopted by the architecture: isolated excursions do not immediately generate alarms, and recovery is only recognised once the indicator rises above the clearance threshold. The resulting output is therefore stable enough for day-to-day monitoring yet still highlights subsystem-specific deterioration patterns that could be relevant for maintenance.

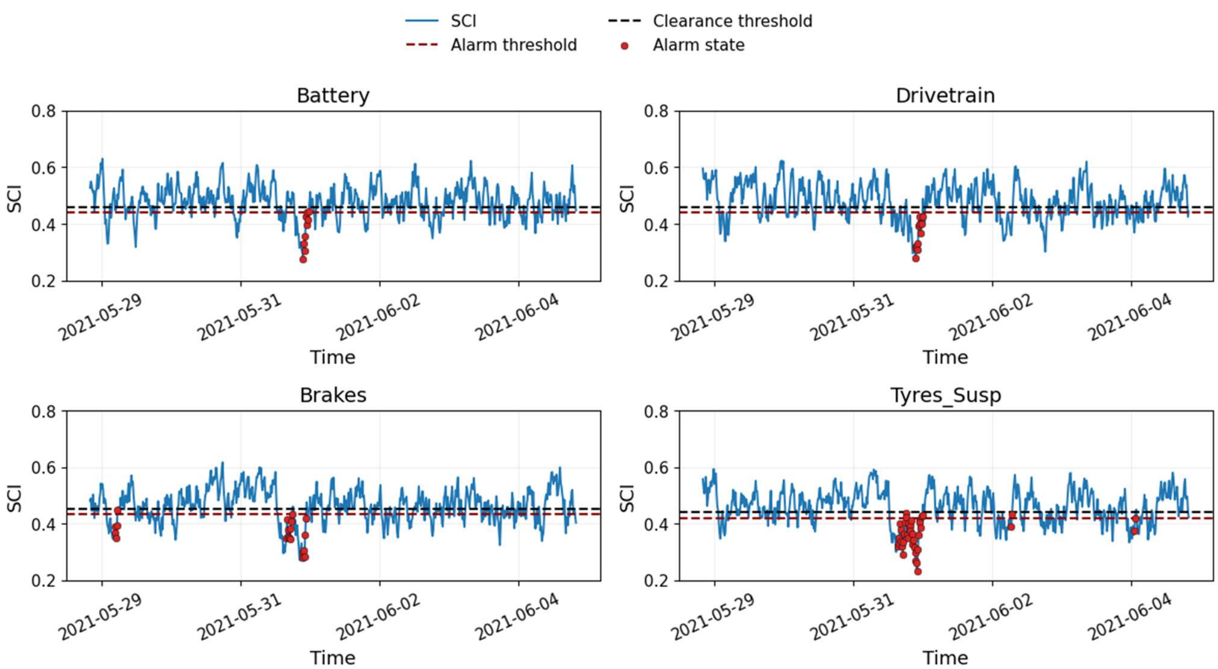

Fig. 2. Subsystem SCI trajectories with frozen thresholds and alarm markers.

A second indication of applicability relates to the burden and manageability of the alerts generated. While a monitoring architecture may be technically feasible, it could still be impractical if it produces excessive, unstable or ambiguous alarms. For this reason, the case study examines the mean number of alarm episodes per day for each subsystem (see Fig. 3), as well as a broader set of operational metrics reported in Table V. In this study, an alarm episode is defined as the start of an alarm period following a non-alarm state and not as the total number of consecutive samples below the threshold. This definition is more meaningful from a maintenance perspective because it reflects how often the monitoring system actually requires attention. Both Fig. 3 and Table V cover the entire case study analysis period. In practical deployments, the same metrics could be recalculated using rolling daily or weekly windows depending on the intended monitoring workflow.

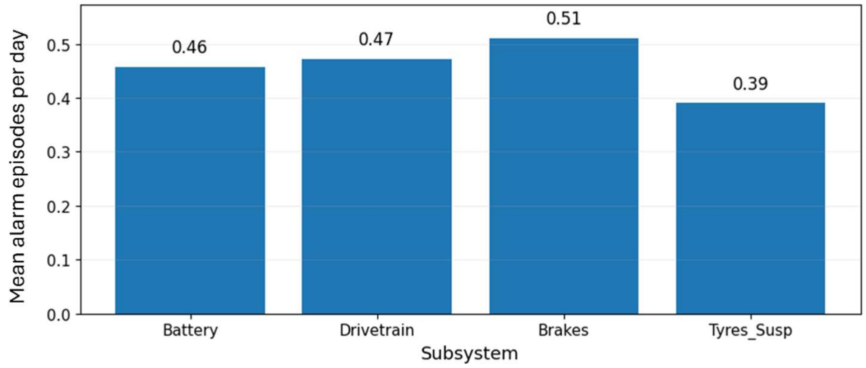

Fig. 3. Mean alarm episodes per day for each subsystem, computed over the full case-study horizon.

As shown in Fig. 3, the alert burden remains relatively low and consistent across subsystems throughout the analysis period, with mean rates of alarm episodes ranging from approximately 0.39 to 0.51 per day. These values are obtained by dividing the total number of alarm activations observed for each subsystem by the corresponding observation period in days. Brakes exhibit the highest alert frequency, tyres/suspension the lowest and battery/drivetrain lie in between. This suggests that the proposed architecture disperses deviations across multiple alert streams without concentrating them in a single dominant stream. Consequently, it provides a differentiated view of subsystem behaviour while maintaining an alert volume that remains manageable in operational terms.

Table V provides a summary of the main operational characteristics of the alerts over the same analysis horizon as Fig. 3. As well as the mean number of alarm episodes per day, it reports the median SCI, the proportion of time spent in the critical state, the mean and maximum durations of alarm episodes, and the median time between consecutive alarm activations. The SCI is presented as the median to provide a more robust description of typical subsystem conditions in the presence of short-lived excursions and alarm-related deviations. Taken together, these metrics suggest that the generated alerts correspond to structured periods of deviation and not sporadic threshold crossings. Mean alarm durations remain at around two hours, maximum durations remain limited and the typical time between events remains well above one day for all subsystems. This is consistent with an alerting logic designed to support observation and prioritisation over the generation of unstable alarms.

TABLE V. MAIN OPERATIONAL ALERT METRICS PER SUBSYSTEM, COMPUTED OVER THE FULL CASE-STUDY HORIZON.

Subsystem	*Median SCI*	*Mean episodes per day*	*Time fraction in critical state (%)*	*Mean alarm duration (h)*	*Max alarm duration (h)*	*Median inter-event time (h)*
Battery	0.474	0.46	4.07	2.13	13.50	34.88
Drivetrain	0.465	0.47	4.06	1.91	9.50	34.25
Brakes	0.472	0.51	4.09	2.02	12.50	34.38
Tyres/Suspension	0.475	0.39	3.29	1.92	9.25	42.50

In line with the scope of this evaluation, these alerts should not be interpreted as confirmed faults or as automatic indications that maintenance is required immediately. Their purpose is to provide an early warning that requires observation, contextual assessment and, where appropriate, further inspection. In practice, some alerts may correspond to transient or easily recoverable deviations, some may be deemed non-actionable, and some may prompt maintenance planning or diagnostic follow-up. Therefore, the objective is not to optimise alarm counts against an external ground truth, which is unavailable in this label-scarce setting, but to verify that the architecture produces outputs that are both interpretable and operationally governable.

Once the alerting outputs have been characterised, it is important to assess their stability under plausible non-ideal conditions, without relying on a specific baseline choice or complete data. Table VI therefore summarises the robustness of the outputs under two moderate perturbation settings: an alternative baseline definition and additional missing data. For baseline variation, the reference SCI is compared with the SCI obtained when the baseline window is set to 20% of the available time series. For missingness stress testing, 10% of the values in the multivariate telemetry matrix are randomly masked, after which the matrix is reprocessed through the same monitoring pipeline. SCI rank correlation is measured using Spearman's rank correlation coefficient, evaluating stability in terms of preserving the overall SCI trajectory ordering over exact pointwise agreement (Stephanou and Varughese, 2021). Values close to one therefore indicate that the overall subsystem monitoring pattern remains essentially unchanged. The remaining quantities report the absolute change in the mean number of alarm episodes per day and the proportion of time spent in the critical state under the 10% missingness condition. This captures the practical effect of the perturbation on the burden of alerts.

The results show that the monitoring outputs remain highly stable under both considered perturbation settings. Under the alternative baseline condition, where the baseline window is set to 20% of the available time series, the Spearman SCI rank correlations are near one for all subsystems. This indicates that the indicator's overall ordering behaviour is preserved. In the case of 10% missingness, Spearman SCI rank correlations remain at least 0.92, while the magnitude of the absolute changes in mean alarm episodes per day and critical-state time fraction remains limited. This suggests that moderate baseline variation and additional missingness do not significantly affect the operational picture produced by the architecture.

TABLE VI. ROBUSTNESS SUMMARY UNDER ALTERNATIVE BASELINE DEFINITION AND ADDITIONAL MISSINGNESS (I.E. MISSING DATA).

Subsystem	*Spearman SCI rank correlation (20% baseline condition)*	*Spearman SCI rank correlation (10% missingness)*	*Absolute change in mean alarm episodes/day (10% missingness)*	*Absolute change in critical state (%) (10% missingness)*
Battery	1.00	0.93	+0.01	+0.15
Drivetrain	0.99	0.92	-0.05	-0.13
Brakes	0.99	0.92	-0.08	-0.38
Tyres/Suspension	0.99	0.92	-0.01	-0.10

V. OPERATIONAL IMPLICATIONS

Section IV demonstrated that the proposed architecture can produce interpretable and consistent outputs in a realistic, label-scarce case study. The following question is whether these outputs are useful in practice. While not aiming for automatic fault diagnosis, the proposed approach organises raw telemetry into early-warning information at the subsystem level that can support monitoring, triage and initial inspection planning.

Compared with simpler alternatives such as fixed sensor thresholds, vehicle-level anomaly scores, and single-channel reconstruction-based monitoring, the proposed architecture is designed to provide more operationally usable outputs. While fixed thresholds are easier to implement, they may be sensitive to context and generate unrelated alerts across multiple variables. Vehicle-level anomaly scores support generic ranking, but they provide limited information about where to begin inspection. Single-channel, reconstruction-based monitoring captures unusual multivariate states, but it can miss deviations that primarily emerge through abnormal short-term dynamics. The proposed architecture combines subsystem-level grouping, reconstruction and forecasting evidence, baseline-referenced calibration and alarm-governance logic to offer a more structured compromise between technical anomaly detection and maintenance usability.

One operational advantage lies in the subsystem-level structure of the outputs. Since one SCI is generated for each monitored subsystem, the user is not limited to a generic indication that the vehicle is deviating from normal behaviour. Instead, the monitoring interface can indicate whether the most significant deviation is primarily associated with the battery, drivetrain, brakes or tyres/suspension. This is important in practice, as inspection and maintenance activities are typically organised around subsystem responsibilities rather than a single, undifferentiated, vehicle-level score.

Unlike the aggregate metrics reported in Section IV, which are computed over the full case-study horizon for evaluation purposes, Fig. 4 provides a dashboard-oriented view based on a rolling window set to the last 24 hours. This reflects how outputs could be monitored on a daily basis. Panel (a) provides an immediate overview of the recent condition of the subsystem by reporting the mean SCI over the last 24 hours. This is computed from the corresponding 15-minute SCI updates together with the associated alarm-on and clearance thresholds. Panel (b) complements this view by reporting the mean short-term SCI trend over the same 24-hour period. This indicates whether recent behaviour is stable, improving (increasing SCI) or deteriorating (decreasing SCI). In the example, none of the subsystems are currently in alarm mode, but panel (a) still allows comparison of how close each subsystem is to the threshold region. The trend lines in Panel (b) show slight negative trends in drivetrain, brakes and tyres/suspension, with tyres/suspension showing the steepest decline. From an operational perspective, this enables the user to transition from a general condition overview to subsystem prioritisation and closer observation within the same interface.

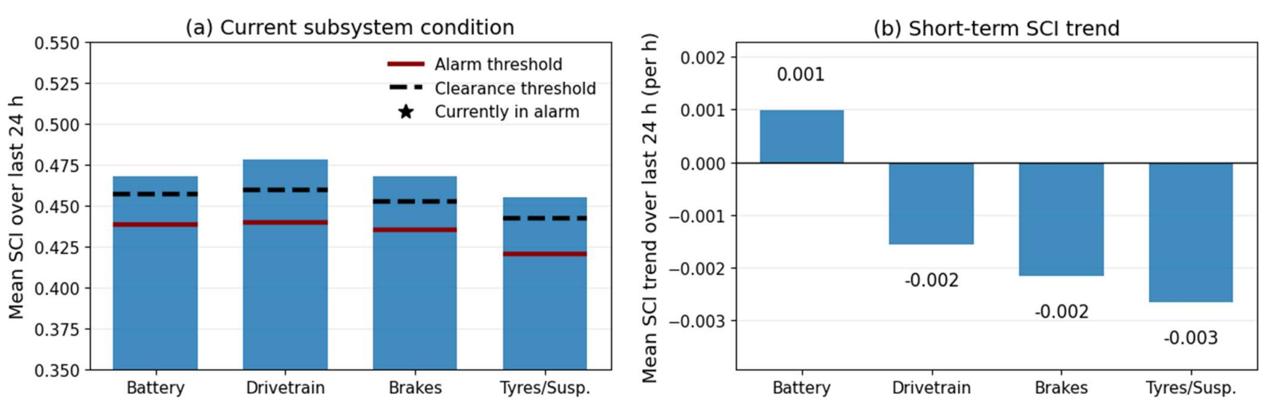

Fig. 4. Example operational interpretation of the monitoring outputs. Panel (a) summarises recent subsystem SCI levels, while panel (b) shows the corresponding short-term SCI trends.

More broadly, the architecture converts generic anomaly signals into structured, subsystem-level monitoring outputs that can be interpreted within an operational context. These outputs can then be used to support the prioritisation of tasks, the planning of inspections, and the decision-making process for maintenance. Its practical value lies not only in signalling that a deviation exists, but also in helping to determine where to look first and which subsystem deserves closer attention. It can also help to establish whether the observed behaviour is stable or progressively deteriorating. In this sense, the dashboard represents an initial step towards providing more prescriptive maintenance support, as persistent alarms and subsystem-specific cues could eventually be linked to recommended inspection or intervention actions.

Table VII supplements Fig. 4 by reporting representative first-inspection cues derived from reason codes. These cues identify the variables most strongly associated with the current subsystem state or with a detected deviation. They should be interpreted as cues to support inspections and not to explain the root cause. They improve operational clarity, reduce ambiguity in alarm handling, and support maintenance triage by indicating where technical assessment could begin. These cues are useful even when no alarm is active, as they help to focus attention on the variables that contribute most to the monitoring output at the subsystem level.

Together, Fig. 4 and Table VII demonstrate how the proposed architecture can serve as a practical maintenance support layer between raw telemetry and engineering judgement. The SCI provides a concise subsystem-level condition summary, the short-term trend adds anticipatory value, and the reason-code-based cues support first-line drill-down. Overall, the result is a monitoring output that is more maintenance-oriented and can be used directly for practical supervision, inspection prioritisation, and supporting first-line maintenance decisions in connected-vehicle settings where labels are scarce.

TABLE VII. REPRESENTATIVE FIRST-INSPECTION CUES DERIVED FROM SUBSYSTEM REASON CODES.

Subsystem	***Current alarm state***	***Suggested first inspection cue***
Battery	No alarm	SoC
Drivetrain	No alarm	Motor vibration
Brakes	No alarm	Regenerative braking efficiency
Tyres/Suspension	No alarm	Load weight

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a subsystem-oriented monitoring architecture for connected vehicles based on unlabelled telemetry. The proposed tool transforms multi-sensor vehicle data into SCIs designed to support maintenance interpretation through bounded condition signals, stable alert logic, and concise first-inspection cues. In this way, the architecture does more than detect deviations: it organises them into dashboard-oriented outputs that can support monitoring, prioritisation, and initial maintenance decision-making at subsystem level, while also providing a basis for future prescriptive-maintenance extensions.

The main contribution of the work lies in showing how established modelling components can be integrated into an operationally meaningful monitoring layer for label-scarce settings, which represents a practically relevant contribution in the field. The EV telemetry case study demonstrated that the proposed architecture preserves subsystem differentiation and generates interpretable, manageable alert behaviour. It also remains stable under moderate baseline variation and additional missing data. These properties are important because they transform the output from a generic anomaly signal into a structured representation that can help to determine where attention should be directed, which subsystem may require escalation and how initial inspection efforts can be prioritised.

Although the present evaluation is based on a single-vehicle dataset, the architecture is designed for fleet-oriented deployment. In this sense, this work does not constitute fleet-level validation, but the definition and initial evaluation of an interpretable, subsystem-level monitoring layer that could support inspection prioritisation, maintenance coordination and more scalable, connected-vehicle supervision. Future work will extend the evaluation to multi-vehicle datasets to assess cross-vehicle comparability and prioritisation strategies at fleet level. When partial maintenance outcomes are available, they can be used to refine alert governance and evaluate the practical usefulness of the generated alerts, eliminating the need for dense failure labels. Further developments will also address refreshing the baseline and operational drift, including the effects of seasonality, ageing-related shifts and sensor recalibration. Finally, the link between alarm patterns and reason-code information and recommended inspection or intervention actions will be explored as a step towards providing more prescriptive maintenance support.

References

Ansari, F., Glawar, R. and Nemeth, T. (2019), “PriMa: a prescriptive maintenance model for cyber-physical production systems”, International Journal of Computer Integrated Manufacturing, Taylor & Francis, Vol. 32 No. 4–5, pp. 482–503.

Benhanifia, A., Cheikh, Z. Ben, Oliveira, P.M., Valente, A. and Lima, J. (2025), “Systematic review of predictive maintenance practices in the manufacturing sector”, Intelligent Systems with Applications, Vol. 26, p. 200501.

Chandola, V., Banerjee, A. and Kumar, V. (2009), “Anomaly detection: A survey”, ACM Computing Surveys (CSUR), ACM New York, NY, USA, Vol. 41 No. 3, pp. 1–58.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014), “Empirical evaluation of gated recurrent neural networks on sequence modeling”, ArXiv Preprint ArXiv:1412.3555.

Giannoulidis, A. and Gounaris, A. (2023), “A context-aware unsupervised predictive maintenance solution for fleet management”, Journal of Intelligent Information Systems, Vol. 60 No. 2, pp. 521–547.

Heger, J., Desai, G. and El Abdine, M.Z. (2020), “Anomaly detection in formed sheet metals using convolutional autoencoders”, Procedia CIRP, Vol. 93, pp. 1281–1285.

Hu, Y., Miao, X., Si, Y., Pan, E. and Zio, E. (2022), "Prognostics and health management: A review from the perspectives of design, development and decision", Reliability Engineering & System Safety, Vol. 217, p. 108063.

Jardine, A.K.S., Lin, D. and Banjevic, D. (2006), "A review on machinery diagnostics and prognostics implementing condition-based maintenance", Mechanical Systems and Signal Processing, Vol. 20 No. 7, pp. 1483–1510.

Kaggle. (2025), "EVIoT-PredictiveMaint Dataset", available at: https://www.kaggle.com/datasets/datasetengineer/eviot-predictivemaint-dataset (accessed 30 December 2025).

Lipu, M.S.H., Hannan, M.A., Hussain, A., Hoque, M.M., Ker, P.J., Saad, M.H.M. and Ayob, A. (2018), "A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations", Journal of Cleaner Production, Elsevier, Vol. 205, pp. 115–133.

Mahale, Y., Kolhar, S. and More, A.S. (2025), "A comprehensive review on artificial intelligence driven predictive maintenance in vehicles: technologies, challenges and future research directions", Discover Applied Sciences, Vol. 7 No. 4, p. 243.

Nunes, P., Santos, J. and Rocha, E. (2023), "Challenges in predictive maintenance – A review", CIRP Journal of Manufacturing Science and Technology, Vol. 40, pp. 53–67.

Resanovic, D., Softic, S. and Balc, N. (2026), "Explainable Anomaly Detection for Predictive Maintenance Using Synthetic Benchmarks and Human-in-the-Loop Labeling BT - Advances in Production Management Systems. Cyber-Physical-Human Production Systems: Human-AI Collaboration and Beyond", in Mizuyama, H., Morinaga, E., Nonaka, T., Kaihara, T., von Cieminski, G. and Romero, D. (Eds.), , Springer Nature Switzerland, Cham, pp. 89–103.

Serradilla, O., Zugasti, E., Rodriguez, J. and Zurutuza, U. (2022), "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects", Applied Intelligence, Springer, Vol. 52 No. 10, pp. 10934–10964.

Severson, K.A., Attia, P.M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M.H., et al. (2019), "Data-driven prediction of battery cycle life before capacity degradation", Nature Energy, Nature Publishing Group UK London, Vol. 4 No. 5, pp. 383–391.

Sharma, A. (2024), "Making electric vehicle batteries safer through better inspection using artificial intelligence and cobots", International Journal of Production Research, Vol. 62 No. 4, pp. 1277–1296.

Stephanou, M. and Varughese, M. (2021), "Sequential estimation of Spearman rank correlation using Hermite series estimators", Journal of Multivariate Analysis, Vol. 186, p. 104783.

Swaroopa Rani, B. and Jatoth, C. (2025), "Deep Learning and Blockchain-Enabled Predictive Maintenance in Electric Vehicles: A Comprehensive Review", Concurrency and Computation: Practice and Experience, Wiley Online Library, Vol. 37 No. 25–26, p. e70298.

Taylor, P., Griffiths, N., Bhalerao, A., Anand, S., Popham, T., Xu, Z. and Gelencser, A. (2016), "Data mining for vehicle telemetry", Applied Artificial Intelligence, Taylor & Francis, Vol. 30 No. 3, pp. 233–256.

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M. and Elger, G. (2021), "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry", Reliability Engineering & System Safety, Vol. 215, p. 107864.

Warton, D.I. (2008), "Penalized normal likelihood and ridge regularization of correlation and covariance matrices", Journal of the American Statistical Association, Taylor & Francis, Vol. 103 No. 481, pp. 340–349.

Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S. and Li, G.P. (2020), "Predictive maintenance in the Industry 4.0: A systematic literature review", Computers & Industrial Engineering, Vol. 150, p. 106889.