Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p24045-cd

Enhancing Automated Warranty Repair Reports Analysis with AI Generative Technologies: Failure Modes Detection in Industrial Vehicles for Reliability Improvement

Fabio Verdinelli

Advanced Reliability Department, CNH Industrial Italy Spa, E-mail: fabio.verdinelli@cnh.com

Mathew Thomas

Advanced Reliability Department, CNH Industrial America LLC, E-mail: mathew.thomas@cnh.com

Abstract: Warranty claims provide valuable information for identifying recurring failures and improving industrial vehicles. This work presents a practical framework that combines community-detection-based clustering and AI-assisted semantic analysis to process large, heterogeneous maintenance reports. The method integrates human-defined labels with AI-generated insights, handling inconsistent terminology and sparse taxonomies to identify robust failure patterns. Thousands of warranty reports per job run were analyzed, producing structured labels and clusters that highlight recurring failure modes, related symptoms, and interactions across different vehicle systems. The approach uses community detection techniques to group similar issues, making patterns easier to interpret than manual review alone. The system is designed for scalability and reproducibility, allowing experts to refine clusters and verify results iteratively. Interactive dashboards enable hierarchical exploration of failures, helping engineers track issues over time and understand cross-system effects. Results show that combining basic statistical signals with semantic information improves the clarity and practical usability of warranty data. The framework helps convert unstructured maintenance reports into structured evidence that engineers can interpret and validate directly. It supports reliability engineering, design improvements, and after-sales operations while remaining adaptable to different datasets and industrial contexts.

Keywords: Warranty data analysis; industrial vehicles; failure modes detection; Jaccard similarity; Louvain clustering; generative AI; reliability engineering; dashboards; label harmonization.

1. Introduction

Industrial and agricultural vehicles operate in varied environments, generating warranty claims that include symptoms, suspected causes, replaced components, and dealer notes. These claims support reliability analysis, guide design validation, and inform after-sales decisions. However, large-scale analysis is complicated by inconsistent wording, irregular abbreviations, and differences in dealer taxonomies. Previous studies show that maintenance and warranty narratives often contain noisy and incomplete textual information that must be standardized before meaningful analysis can be performed.

Research on warranty and maintenance text can be grouped into three main areas. First, traditional text-mining approaches rely on bag-of-words or TF-IDF representations, which are sensitive to lexical variation, do not capture semantic similarity, and often suffer from feature sparsity in short technical narratives such as warranty comments [1]. Second, supervised machine-learning approaches have been explored for classifying maintenance events, but they require large, consistently annotated datasets and their performance depends strongly on domain-specific training data, which limits their reusability across heterogeneous warranty sources [2]. In parallel, unsupervised semantic approaches use AI language models to convert text into numerical representations of meaning, enabling similar issues to be grouped even when phrasing differs [3]. However, these methods remain document-centric and only loosely connected to the structured technical fields used in reliability workflows. For example, they often merge unrelated issues described with similar wording; they may group an air-conditioning compressor with a turbo compressor despite belonging to different systems. In the same way, generic terms associated with fluid leakages tend to produce very large, poorly structured clusters where hydraulic, pneumatic and lubrication problems are mixed simply because they share words such as “leak”, “loss” or “pressure drop”. Third, recent work highlights the need for domain-specific structuring: frameworks such as Technical Language Processing emphasize that engineering and maintenance text requires normalization, consistent annotation, and controlled vocabularies to support reliable downstream analysis [4].

This study introduces a label-centric clustering strategy that operates on technical and semantic labels rather than raw text. Human-defined technical fields (Component Group, Component Code, Causal Part) are combined with AI-derived text labels extracted from dealer comments (Failure Mode, Cause, Effect) to form interpretable groups of related failure modes. Beyond row-level extraction, the method produces AI-generated cluster descriptions that summarize the communities identified by the clustering process, offering concise, engineering-ready explanations of recurring patterns. This formulation reduces the impact of wording variation, harmonizes dealer-specific terminology, and keeps cluster structures aligned with operational fields familiar to engineering teams.

Overall, the proposed approach contributes a label-centric formulation that combines structured technical fields with text-derived labels and AI-generated summaries of each cluster, together with a graph-based community-detection process that yields interpretable and auditable groups of related failure modes.

1. Business Need and Challenges

Manual review of warranty claims and field failure data is a labor-intensive bottleneck that hampers quality

improvement cycles. Engineers must manually sift through unstructured dealer comments alongside semi-structured technical fields (such as causal part numbers, component groups, and component codes) to identify patterns and verify root causes. The inherent variability in how technicians describe failures introduces significant ambiguity: one dealer might log a symptom as "won't start," while another records "no crank condition" for the same underlying issue. This linguistic inconsistency forces analysts to invest extra time aligning terms before they can even begin data aggregation.

The problem is compounded by inconsistent labeling and vocabulary across different dealers, service networks, and legacy systems. Identical failure modes may be classified under disparate descriptors, which dilute the statistical signal needed to detect emerging trends and complicate cross-system aggregation. Without a unified taxonomy, even straightforward tasks like tracking the prevalence of a specific defect require manual harmonization of data from multiple sources. Three critical needs emerge from this fragmented landscape:

Faster design feedback: Engineering teams require structured, automated summaries that surface recurring issues and quantify their frequency, enabling rapid iteration on design improvements without waiting weeks for manual reports.

Better root cause analysis: Effective problem-solving depends on systematically linking observable symptoms (what failed) to underlying failure mechanics (why it failed) and downstream effects (impact on vehicle performance or safety). Current manual processes struggle to maintain this traceability.

Improved after-sales support: Service organizations need consistent taxonomies and concise claim briefs that allow technicians to quickly diagnose issues, access relevant repair procedures, and ensure customers receive accurate, timely resolutions. Standardized language also improves communication between field personnel and engineering, closing the feedback loop more effectively.

2. Data and Output Scope

The dataset includes warranty claims from industrial and agricultural vehicles, covering different product types such as tractors, harvesters, and construction machines. These vehicles have been in operation over the last 2–3 years, and warranty coverage is typically one year. The largest dataset contained around 20 thousand claims, which was used to test the application for scale, robustness, and general applicability across different fleets. Each claim contains free-text dealer notes and structured technical fields such as Causal Part, Component Group, and Component Code. Outputs: The process generates AI-based labels (System, Component, Failure Mode, Failure Cause, Failure Effect), short claim summaries, clusters of failure modes with AI context, and interactive dashboards for analysis. Label provenance: Core technical labels (Component Group, Component Code, Causal Part) are defined by service technicians from existing taxonomies. Auxiliary semantic labels (System, Component, Failure Mode, Failure Cause, Failure Effect) are derived from dealer comments and structured fields. Only a subset of auxiliary labels (Failure Mode, Cause, Effect) is used to build the similarity graph; System and Component labels are used for summarization, filtering, and validation. Dataset characteristics: Claims vary in length, terminology, and detail, reflecting real-world reporting variability. Testing the approach across several fleets helped confirm the flow's robustness on different types of data.

3. End-to-End Pipeline

Batched ingestion: Claims data is ingested through cloud-based data pipelines that process records in batches at regular intervals. This approach ensures high throughput and scalability while providing a buffer against pipeline instability. Because the ingestion pipeline can be unreliable and prone to breaking, batched processing allows the system to recover from failures, retry failed batches, and maintain data integrity without losing records.

Automated AI labeling: For each individual claim, a generative AI model analyzes the unstructured text (dealer comments, technician notes) and structured fields (part codes, component identifiers) to produce row-level labels and a concise natural-language brief. The brief captures four key dimensions: the affected system (e.g., powertrain, HVAC), the specific component (e.g., fuel pump, compressor clutch), the failure mechanics (e.g., seal degradation, electrical short), and the downstream effect (e.g., no-start condition, reduced cooling performance). Component Group, Component Code, and Causal Part are human-defined technical fields, while System, Component, Failure Mode, Cause, and Effect are AI-generated.

Label grouping (sparsity reduction): Raw AI-generated labels often exhibit variability due to synonyms, abbreviations, and minor linguistic differences. To address this, the AI first analyzes the full set of generated labels across all claims and creates a reduced set of canonical categories that consolidate semantically similar terms. For example, variations like "O-ring", "seal," and "gasket" are mapped to a single unified category when they refer to the same functional part. Once these standard categories are set up, the AI performs a second pass over the dataset, assigning each claim's original labels to the appropriate canonical categories on a line-by-line basis. This two-stage process reduces label sparsity, harmonizes disparate taxonomies across dealers and systems, and ensures that statistically similar failure modes are not artificially fragmented by minor wording differences. The grouping does not alter core human-defined fields; it harmonizes only semantic variants so that auxiliary AI labels better align with the existing technical taxonomy.

With harmonized labels in place, the flow builds a label–label similarity graph and applies a community detection algorithm to obtain interpretable groups of labels

representing failure-mode bundles. Each claim is then mapped to one or more communities via its labels, and generative AI synthesizes a concise description for each community, summarizing common characteristics, listing main failure mechanisms, effects and possible root causes. This hybrid approach combines graph-based clustering with natural language summaries that support engineering interpretation.

Results are finally surfaced through a web-based dashboard that gives engineers and analysts intuitive access to the processed data. Users can filter claims by vehicle model, date range, component category, or failure severity. They can drill down into specific clusters to examine constituent claims, review label distributions, and trace patterns back to individual warranty records. Visualizations such as histograms and time-series support rapid pattern recognition. Results can include standard warranty metrics such as average cost or number of claims/relative frequency per vehicle. The dashboard also enables collaborative workflows: teams can annotate findings, flag clusters for further investigation, export summaries for design reviews, and share insights with after-sales or product development teams to drive corrective actions continuous improvement.

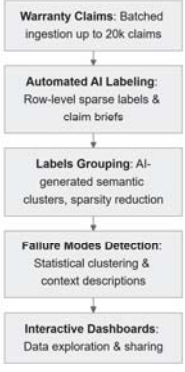

Fig. 1. End-to-End pipeline: Batched ingestion → AI labeling → labels grouping → Failure modes detection (Statistical + AI) → Interactive dashboards.

4. Failure Modes Detection Pipeline

To identify coherent groups of failure-related labels, each label is a node in a weighted graph. Edges represent how strongly labels are related across claims and technical context. Similarity is measured with a set-based metric (see Eq. (1), Jaccard similarity [5]), which allows even rare or unusual label co-occurrences to create meaningful connections. For label i, let A_i be the set of claims where i occurs. Pairwise Jaccard similarities are computed between labels (not between claims) on a label–label graph restricted to core human-defined fields (Component Group, Component Code, Causal Part) and the auxiliary roles used for clustering (Failure Mode, Cause, and Effect).

In this way, the graph captures relationships across the dataset, allowing patterns to emerge naturally. Once the graph is built, communities are detected using the Louvain community detection algorithm [6]. This method searches for groups of labels that are more tightly connected to each other than to the rest of the graph. The measure used, modularity (see Eq. (2), [6]), evaluates how much a group's internal connections exceed what would be expected by chance, favoring clusters that are internally dense and externally sparse. Here, m denotes half of the total edge weight in the graph and is used to normalize modularity values (see Eq. (3)).

$$w_{ij} = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \tag{1}$$

$$Q = \Sigma_C \left[\frac{w_{in(C)}}{2m} - \left(\frac{w_{tot(C)}}{2m} \right)^2 \right] \tag{2}$$

$$m = \frac{\sum_{ij=1}^{n} wij}{2} \tag{3}$$

The algorithm alternates between local node moves, where each label is placed in the group that maximizes modularity, and community aggregation, where detected groups are collapsed into super-nodes to form a hierarchy. This process allows the identification of both fine-grained clusters (capturing specific failure patterns) and broader clusters (revealing general trends).

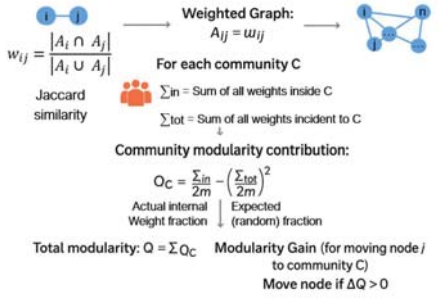

Fig. 2. Louvain Community Detection Algorithm: Communities are detected on a weighted similarity graph. The algorithm iteratively maximizes modularity and aggregates nodes hierarchically.

To ensure that the clusters are coherent and technically meaningful, domain-informed weighting and similarity thresholds are applied. In addition, a top-k pruning step is used to reduce graph complexity: for each core label (Component Group, Component Code, Causal Part), only the k strongest edges toward auxiliary labels (Failure Mode, Cause, Effect) are retained, while weaker connections are removed. This focuses the graph on the most significant semantic relationships, easing the

detection of interpretable label clusters. Fig. 3 describes the major logical steps of the Failure Mode Detection pipeline.

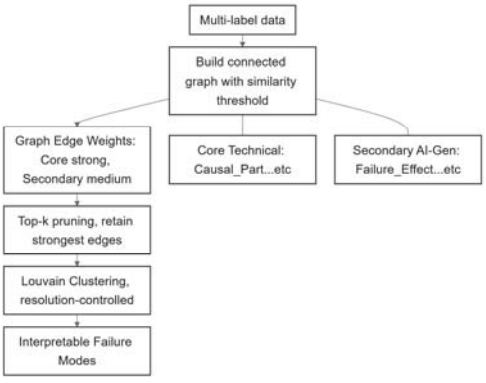

Fig. 3. Failure Mode Detection pipeline. The pipeline processes multi-label maintenance data, applying row weights to emphasize frequent cases. Labels are weighted by semantic role: core labels (Causal Part, Component Code, Component Group) carry high influence, secondary labels (Failure Mode, Failure Cause, Failure Effect) moderate influence. Edges are pruned using a top-k strategy from core to secondary labels to keep the graph interpretable. Louvain community detection, then AI-generated descriptions and interpretable clusters, highlighting recurring failure patterns.

The clustering process is guided by two key parameters: the similarity threshold, which filters out weak or spurious label relationships, ensuring that only meaningful connections contribute to the graph structure, and the Louvain resolution, which controls the granularity of the detected clusters. Together, these parameters steer the algorithm to capture both fine-grained failure modes and broader failure trends, producing clusters that are coherent, interpretable, and reflective of the underlying technical context, providing a reliable basis for analyzing recurring failure modes.

In practical terms, each Louvain community corresponds to a bundle of labels that frequently co-occur across claims and therefore provides a data-driven approximation of a failure mode together with its associated symptoms, causes and effects. These communities reveal stable technical patterns that are often difficult to detect through manual review or through document-level clustering. After the communities are formed, AI is used to generate concise descriptions of each cluster, providing engineering-ready summaries of the dominant failure patterns and the relationships among labels.

5. Solution Implementation

The system is deployed in a cloud environment built around a centralized data lake. Raw claims data, intermediate artifacts, and derived outputs (labels, clusters, and summaries) are stored as versioned tables, enabling full traceability across processing runs. This separation allows raw inputs to remain immutable while supporting iterative enrichment and re-processing. Batch processing jobs orchestrate the core analytical pipeline. These jobs handle large-scale AI-based labeling, multi-dimensional clustering, and post-processing steps, operating on controlled batches to ensure consistency across claims and runs. Each execution produces structured outputs that are published back to the data lake, making results immediately available for downstream analysis and querying. On top of the data lake, an application layer provides access to both operational and analytical views. A web application enables users to manage runs, inspect label and cluster distributions, and perform claim-level drilldowns combining original text, technical fields, and contextual metadata. The platform is designed to support collaboration and governance. Data scientists maintain and evolve the processing pipeline and prompting logic, while reliability professionals validate clusters, interpret patterns, and translate them into engineering actions. Their feedback is systematically captured and reintegrated into subsequent runs, closing the loop between automated analysis and domain expertise.

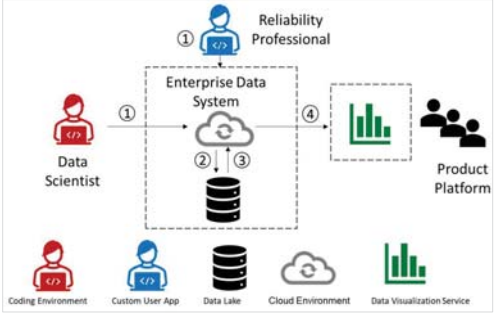

Fig. 4. Solution architecture: cloud data lake and coding environment; web app UI; dashboard; roles (data scientist and reliability professional).

6. Exploration and Visualization

Dashboards translate clustering results into an interactive exploration layer that supports both analytical insight and operational engineering workflows.

Each failure mode cluster corresponds to a community of labels discovered on the label–label graph (edge weight: Jaccard on claim sets). As illustrated in Fig. 5, users can start from macro-level views, such as dominant failure mode clusters ranked by claim frequency, and progressively drill down into the associated technical labels and underlying warranty records. This top-down navigation enables engineers to quickly identify recurring issues and assess their relative impact before inspecting the supporting evidence at the claim level.

The exploration workflow is complemented by AI-generated semantic summaries that describe each cluster in terms of component, failure mode, failure cause, and failure effect. Fig. 6 highlights how these concise

descriptions provide immediate contextual understanding, reducing the effort required to interpret heterogeneous and inconsistently worded claims.

By linking aggregated statistics with structured semantic explanations, the dashboards preserve interpretability while supporting rapid triage, failure cause investigation, and structured design feedback loops.

Interactive visual filters allow results to be sliced by system, component, time window, or usage context, enabling flexible analysis across different engineering perspectives. In practice, this exploration layer also acts as a validation mechanism: inconsistencies, edge cases, or shifts in failure behavior can be detected visually and fed back into prompt refinement and iterative re-clustering, closing the loop between analysis, expert judgment, and model evolution.

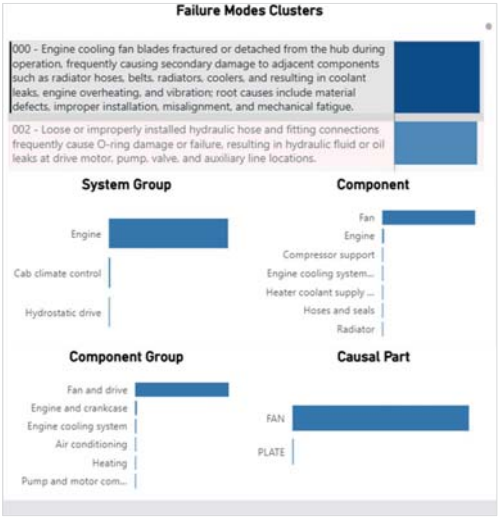

Fig. 5. Dashboard view showing an example of failure modes clusters sorted by claim frequency with related technical labels. Figure is illustrative and does not indicate product-specific or commercially sensitive information.

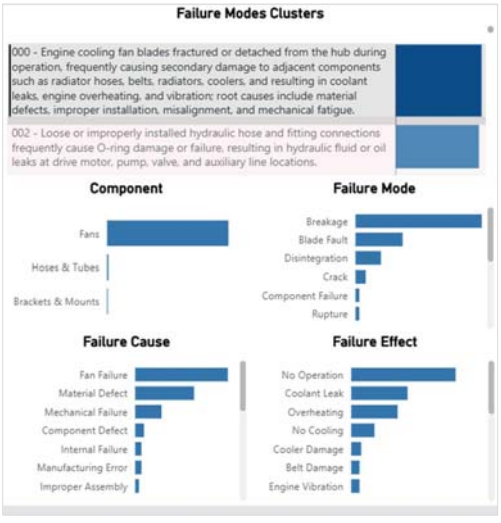

Fig. 6. Dashboard view highlighting AI-generated cluster descriptions: Component, Failure mode, Failure cause, Failure effect. Figure is illustrative and does not indicate product-specific or commercially sensitive information.

7. Results

The hybrid approach produced clusters that were technically precise and straightforward to interpret, giving reliability and design teams a clear view of recurring issues and their context. Variants in wording across dealers and systems were consolidated into a consistent failure-mode taxonomy, enabling reliable aggregation across different operational settings. The clusters also highlighted chains of related failures, for example a seal failure progressing to hydraulic pressure loss and then to clutch lock-up, providing a coherent picture of how symptoms, causes and effects are connected.

Prompting supported several stages of the process. At first it helped identify and group similar labels, allowing coherent clusters to emerge even when the underlying data was sparse or heterogeneous. As the work progressed, prompts were adjusted with expert input to reduce ambiguity, maintain consistency and set an appropriate level of detail. This iterative interaction between automated steps and expert review helped maintain clarity across thousands of claims. The approach also scaled well, processing large batches in roughly one to three hours while preserving the interpretability of the results. By combining structured prompting, targeted expert feedback and automated analysis, the method produced insights that support both diagnostic activities and preventive actions.

8. Discussion

The clusters obtained from the label-centric approach show how symptoms, causes, effects and technical fields tend to recur together across warranty claims. This organization

highlights failure patterns more clearly than document-level methods, which often merge unrelated issues that happen to be described with similar wording, as observed in our previous work based on document-level topic modelling [7]. Working directly with labels also makes the groups easier to relate to the technical fields used by engineers, reducing the risk of mixing heterogeneous failures and improving consistency during analysis. In other words, the method separates cases more reliably and produces clusters that align with existing component taxonomies. The short AI-generated descriptions help summarize the content of each cluster, making it easier to understand the main failure modes without reviewing individual claims. Recent work has explored the use of large language models fine-tuned on annotated maintenance data to classify failure modes [8], but these approaches require structured training sets and aim at code assignment rather than the formation of interpretable groups. The method presented here instead focuses on creating coherent, auditable label clusters without relying on supervised training.

The method also offers flexibility. Parameters such as the similarity threshold, the top-k pruning step and the resolution of the community-detection stage can be adjusted to control the level of detail, depending on whether the analysis aims to identify specific mechanisms or broader trends. This makes the approach suitable for both targeted investigations and higher-level reviews.

Some limitations remain. Text-derived labels depend on how clearly issues are described in dealer notes, and ambiguous wording may still require expert judgement. Cluster boundaries can change when data distributions shift or when parameters are tuned differently, meaning that domain review remains an important part of the process. Despite these considerations, the method provides a practical and interpretable way to organize large volumes of warranty claims and extract technical insight that supports reliability assessments, design feedback and product improvement.

9. Conclusion and Future Work

The approach provides an interpretable, industry-ready framework for automated warranty analysis in industrial vehicles. By organizing complex and heterogeneous claims into clear patterns, it transforms raw data into actionable insights for reliability and design teams, revealing failure trends and cross-system dependencies. The pipeline emphasizes reproducibility, transparency, and usability, making it suitable for enterprise deployment and adaptable to evolving datasets. Its modular design supports iterative improvement through expert feedback and allows integration of additional data sources as needed. A natural next step is to build validated clusters as a foundation for predictive models. Clustered failure modes and components can be used to create life models, such as Weibull distributions, allowing estimation of component lifetimes, prediction of failure probabilities, and risk quantification. This translates qualitative insights from warranty data into measurable reliability indicators that support maintenance planning and design decisions. Beyond classical life modeling, clusters can guide other predictive techniques, including regression and machine learning models. Coherent, harmonized failure categories reduce noise in the input data and improve model robustness, especially when historical data are sparse or inconsistent. Integrating multi-modal data, such as telemetry, sensor readings, and diagnostic codes, can enrich clusters, strengthen causal interpretation, and improve predictive performance. Active learning strategies can keep the system aligned with new data and expert knowledge, using feedback on cluster assignments and grouping thresholds to refine both clustering and predictive models. This feedback loop helps the pipeline evolve from simple descriptive analysis to a tool that supports forward-looking reliability decisions.

References

[1] Stenström, C., Aljumaili, M., & Parida, A. (2015). Natural Language Processing of Maintenance Records Data. Division of Operation and Maintenance Engineering, Luleå University of Technology.

[2] Mayhew, P. J., Ihshaish, H., Deza, I., & Del Amo, A. (2023). Maintenance Automation Using Deep Learning Methods: A Case Study from the Aerospace Industry. ICANN 2023, LNCS 14263, 295–307. https://doi.org/10.1007/978-3-031-44204-9_25

[3] Grootendorst, M. (2022). BERTopic: Neural Topic Modeling with a Class-Based TF-IDF Procedure. arXiv:2203.05794.

[4] Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2020). Technical Language Processing: Unlocking Maintenance Knowledge. National Institute of Standards and Technology (NIST).

[5] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles, 37, 547–579.

[6] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.

[7] Verdinelli, F., & Thomas, M. (2024). Boosting Warranty Data Analysis with Natural Language AI Algorithms. Proceedings of the Annual Reliability and Maintainability Symposium (RAMS), Albuquerque, NM, 1–6. IEEE. https://doi.org/10.1109/RAMS51492.2024.10457679

[8] Stewart, M., Hodkiewicz, M., & Li, Z. (2023). AI-Based Failure Mode Detection from Maintenance Reports. Reliability Engineering & System Safety, 234, 108242.