Data-Driven Synthesis of Stochastic Fault Trees for Proactive Maintenance of Railway Vehicles

Laura Carnevali[1], Alessandro Fantechi[1], Gloria Gori[1(✉)], Denis Vreshtazi[1], Alessandro Borselli[2], Maria Rosaria Cefaloni[2], and Lucio Rota[2]

[1] DINFO, University of Florence, Florence, Italy
{laura.carnevali,alessandro.fantechi,gloria.gori}@unifi.it, denis.vreshtazi@edu.unifi.it
[2] Trenord s.r.l., Milan, Italy
{alessandro.borselli,mariarosaria.cefaloni,lucio.rota}@trenord.it

Abstract. The spreading of sensor technologies has enabled railway operators to collect increasing amounts of granular data on relevant events of components and systems of railway vehicles and infrastructure, presenting unprecedented opportunities to develop predictive failure models. Our research introduces a novel methodology for synthesizing stochastic fault tree models by strategically integrating extensive diagnostic data logs, maintenance records, and domain-specific knowledge to predict component and system-level reliability dynamics. To demonstrate the potential of the approach, we apply it to the traction control unit of a fleet of regional passenger trains, showing a scalable framework for predictive failure assessment across diverse railway vehicle configurations. By leveraging existing diagnostic infrastructure without requiring additional sensor investments, our approach represents a pathway from reactive diagnostic practices to proactive maintenance strategies.

1 Introduction

Synthesizing maintenance strategies in complex cyber-physical systems, particularly in transportation systems, represents a critical challenge at the intersection of reliability engineering and operational economics. The railway industry exemplifies this challenge, as it faces a striking contrast in maintenance capabilities across its fleet generations. Modern high-speed trains are equipped with sophisticated sensor networks that enable comprehensive condition monitoring and predictive maintenance (PdM) features. However, a significant fraction of the operating fleet consists of older vehicles that rely primarily on traditional time-based maintenance plans. While these legacy vehicles may already incorporate multiple basic diagnostic functions for fault detection and reporting, they lack advanced predictive capabilities, despite being more prone to failures and requiring more frequent corrective maintenance interventions [4,14]. This technological

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
A. Remke and B. Steffen (Eds.): FMICS 2025, LNCS 16040, pp. 162–181, 2026.
https://doi.org/10.1007/978-3-032-00942-5_9

gap in railway fleets creates a compelling opportunity for innovation. The older vehicles are typically equipped with diagnostic sensors (monitoring vibration, temperature, electrical current, mechanical stress, etc.), primarily used to help maintenance engineers or drivers to detect anomalies and trigger maintenance alerts. However, data collected by these sensors also contain valuable patterns and trends that, if properly analyzed, could predict impending failures [14]. The challenge lies in developing methodologies that can effectively leverage this existing diagnostic infrastructure for predictive purposes without requiring further costs for sensor network upgrades. Furthermore, when fault-to-failure propagations (i.e., errors propagating in components up to cause their failure) and failure-to-fault propagations (i.e., component failures acting as external faults for other components) have probabilistic characterization in time, quantitative evaluation of stochastic models of the system failure logic enables derivation of metrics of dependability [18,29,31], supporting early validation of design choices and development of predictive analytics for proactive fault management [30].

Fault Tree Analysis (FTA) has long been established as a reliable method for analyzing system failures in various domains. Its structured approach to mapping failure pathways and understanding component interdependencies makes it particularly suitable for complex systems like the ones onboard railway vehicles. While FTA has traditionally been used in system design and reliability analysis, its application in real-time PdM represents a possible opportunity. The incorporation of dynamic probability assessment into Fault Trees (FTs), especially using real-time sensor data, offers a promising perspective for enhancing their utility in operational contexts.

This paper presents a novel application of FTA to bridge the said gap between the available diagnostic infrastructure and the lack of advanced predictive capabilities, by exploiting existing sensor networks in railway vehicles. Our approach leverages FaultFlow [7,8,26], an open-source library developed by our research group at the University of Florence, to analyze component and system failures, transforming traditional FTs into dynamic predictive models, continuously updated with probability assessments derived from operational sensor data. Given the hierarchical nature of railway systems, where train-level reliability depends on coach-level system reliability, our methodology employs a dual-model approach: FTA for detailed coach-level analysis of system failures and Reliability Block Diagrams (RBDs) for train-level reliability assessment considering system redundancy and operational constraints. The proposed methodology offers several advantages:

1. Cost-effectiveness by using of existing basic diagnostic infrastructure.
2. Integration with established diagnostic systems and maintenance protocols.
3. Transparent decision-making processes based on well-understood FT models.
4. Applicability to legacy railway vehicles with no extensive sensor upgrades.

We applied the approach to a railway case study provided by Trenord s.r.l., predicting the reliability of Traction Control Units (TCUs) on legacy vehicles. In particular, the key scientific contributions of this work include:

- Converting raw diagnostic data into stochastic models, providing an artifact[1] to replicate the experimental results, available under the AGPLv3 licence.
- Validating the feasibility of using FTA for real-time failure prediction.
- Showing how to apply the approach in a case study using real world data.

The remainder of this paper is organized as follows: Sect. 2 provides some related works, Sect. 3 discusses the background, Sect. 4 describes the methodology for fault detection by adapting diagnostic sensor data to FT-based predictive model. The problem definition and case study are presented in Sect. 6. Section 7 presents and discusses the results and their implications. Finally, Sect. 8 concludes the paper pointing out directions for future research.

2 Related Work

Fault detection and diagnosis, as well as PdM, have been extensively explored for railway systems, spanning various approaches from data-driven methods to model-based techniques. Several data-driven methods have demonstrated effectiveness for real-time fault detection and diagnosis in traction systems. Liu et al. [20] applied deep Principal Component Analysis (PCA) for detecting incipient faults in electrical drives, showing significant improvements over traditional methods in terms of early detection capabilities. Similarly, Chen et al. [12] developed adaptive observers to estimate system states and detect sensor faults in traction systems of high-speed trains, achieving robust performance even under variable operating conditions.

Beyond traction systems, PdM techniques have been successfully applied to other critical train components, with specific focus on door systems which frequently experience operational issues affecting service availability. When multiple doors are deactivated due to system issues, service interruptions become inevitable, highlighting the importance of early anomaly detection. Ribeiro et al. [28] employed statistical anomaly detection techniques to predict failures in automatic door systems, while Wang et al. [35] exploited sequential pattern mining to identify abnormal operational signatures in door functioning data. Other critical components have also benefited from data-driven approaches. Davari et al. [14] implemented a Sparse Autoencoder (SAE) network for early failure detection in Air Production Units (APUs), analyzing both analog and digital sensor data to identify anomalies indicative of potential air leakage problems or other failure modes before they manifest as critical issues.

For system-level reliability analysis, various FT analysis tools have been developed with varying capabilities and modeling approaches. Several tools focus on Dynamic Fault Trees (DFTs) [29], which offer enhanced expressivity by modeling dependencies among component behaviors, including dependent events, spare components, and different operational modes, e.g., DFTCalc [1], SAFEST [34], DFTRES [6], and SHyFTOO [13]. Other notable contributions include

[1] https://doi.org/10.5281/zenodo.15613737.

DFTSim [5], RAATS [22], MatCarloRE [21], and RADYBAN [23]. To manage the increased complexity in analyzing the underlying stochastic processes, these tools either constrain duration distributions within the Markovian setting or employ simulation-based solution methods. More general-purpose tools for quantitative evaluation of stochastic models (not specific for dependability evaluation) include SHARPE [32] (for generalized stochastic Petri nets), CPN IDE [33] (for colored Petri nets), TimeNET [36] (for deterministic and stochastic Petri nets), Möbius [15] (for various formalisms including stochastic activity networks), and ORIS [25] (for stochastic time Petri nets). Other tools such as LIFT [24] have addressed learning of static FTs from observed data, with focus on the FT structure rather than on the distribution of the time to FT events.

To perform FTA, we selected the FaultFlow library [7,8,26], which performs complete state space analysis and derives importance measures of faults. Specifically, the Birnbaum measure estimates the impact of the occurrence of a fault on the system time-to-failure Cumulative Distribution Function (CDF), while the Fussell-Vesely measure estimates such impact by taking into account the occurrence probability of any minimal cut set (i.e., minimal combination of faults that induces the system failure) containing the fault. Overall, these measures provide invaluable insights into the contribution of each subsystem to system-level failures, enabling more targeted maintenance strategies.

The considered case study presents train-level reliability where coach redundancy and operational constraints must be considered, which can be easily represented with standard combinatorial reliability models. We exploited RBDs to analyze the train reliability through librbd [9], an efficient open-source library that supports the numerical computation of the reliability curve for all RBD basic blocks.

Particularly relevant to our work is the research by Ferdous et al. [16], who developed data-driven machine learning approaches to predict faults in traction control units of legacy trains. However, their approach significantly differs from ours in two key aspects, i.e., it operates at the train level rather than the more granular coach level, and it relies predominantly on non-explainable black-box machine learning systems In contrast, our methodology leverages domain knowledge to create an explainable framework that operates at the coach level, enabling more precise fault detection and diagnosis. Our approach aligns with recent trends in quantitative dependability evaluation of train control systems in presence of uncertainty, as surveyed by Carnevali et al. [11], while providing the flexibility needed for modern heterogeneous train compositions.

3 Background

3.1 FaultFlow: Model-Driven Dependability Evaluation

FaultFlow [7,8,26] is a Java library for quantitative evaluation of dependability of component-based systems, leveraging a Model-Driven Engineering (MDE) approach to analyze complex failure behaviors. FaultFlow models the hierarchical structure of the system and the behavior of fault propagations within

it, considering both intra-component propagations from a component fault to its failure, and inter-component propagations where a component failure comprises a fault of a higher-level component. In particular, the system structure is specified using a SysML Block Definition Diagram (BDD), while the system failure logic is modeled by a Stochastic Static Fault Tree (SSFT) made of: leaf nodes modeling internal faults; logical gates modeling conditions that activate propagation of a combination of faults into a component failure, i.e., AND (the output event occurs if all input events occur), OR (the output event occurs if any input event occurs), and VOT(k/N) (the output event occurs if at least k of the N input events occur); and, propagation nodes modeling durations of propagations.

The timing of fault occurrences and fault propagations is characterized by non-Markovian Probability Density Functions (PDFs) in the class of Expolynomial functions, potentially with bounded support, allowing for flexible fitting of analytical distributions from statistical data. FaultFlow calculates the Cumulative Distribution Function (CDF) of the time to failure for any failure defined in the SSFT, including the top-level system failure. If the SSFT does not include repeated events, FaultFlow also derives importance measures of faults, characterizing how each fault contributes to a system failure over time.

The typical workflow in FaultFlow consists of the following steps and model-to-model transformations: *i*) the metamodel instance of the system can be automatically created by parsing a JSON file encoding the BDD and the SSFT or, alternatively, it can be programmatically created using the FaultFlow API; *ii*) the metamodel instance is stored in a database; *iii*) for a specific failure mode, the metamodel instance can be transformed into a Stochastic Time Petri Net (STPN), which an be analyzed by the Sirio library of the ORIS tool [25], or into extended UML statecharts termed Hierarchical Semi-Markov Processes with parallel regions (HSMPs) [3,17], which can be analyzed by the Pyramis library [10,27]; *iv*) both stochastic analyses yield the time-to-failure CDF.

In the MDE perspective, FaultFlow provides the following advantages:

- High-level modeling: BDDs and SSFTs facilitate comprehension of complex models with respect to STPNs and UML statecharts, also by domain experts.
- Support for non-Markovian duration distributions: FaultFlow provides flexibility and accuracy in representing real-world systems compared to tools limited to exponential distributions, fitting only the mean value of observed statistics. Specifically, FaultFlow accepts any analytic form in the class of exponomial functions [32] (defined as the sum of products of exponential and polynomial terms), with the same representation over the entire domain or piecewise-defined over multiple sub-domains. If the SSFT does not include repeated events, FaultFlow also accepts any distribution in numerical form.
- Automated analysis: FaultFlow automates the transformation of BDDs and SSFTs into STPNs and HSMPs, and their subsequent stochastic analysis.
- Open-source availability: FaultFlow usage and extension are encouraged.

3.2 Librbd: Highly Effective Hierarchical Reliability Analysis

The `librbd` C library [9,19] supports the evaluation of redundancy behavior of complex systems by performing the analysis of Reliability Block Diagrams (RBDs). Specifically, `librbd` performs numerical evaluation of reliability for compositions of RBD basic blocks, providing computational efficiency, multi-platform support, and open-source availability under the AGPLv3 licence. The library can be effectively used in combination with FaultFlow in redundancy analysis scenarios, exploiting the time-to-failure CDF of system components derived by FaultFlow.

4 Process Description

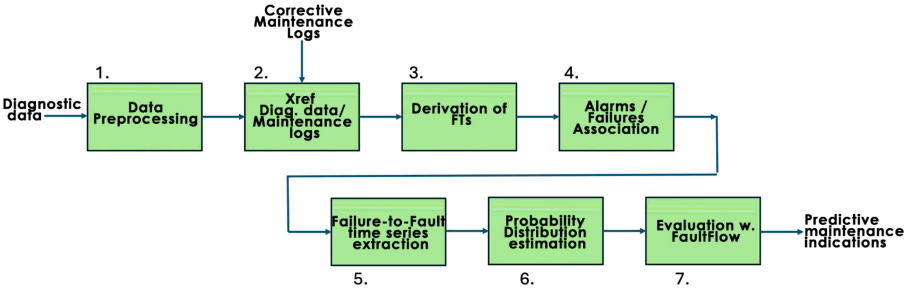

Fig. 1. Predictive maintenance workflow.

The proposed process of elaborating diagnostic data to establish a PdM system involves several methodological steps, each critical to ensure the reliability and accuracy of the predictive framework. In the following sections we describe step by step the workflow shown in Fig. 1. The process has been shaped over the railway case study at hand: it is however presented in rather general terms, in an effort to define a generic data-driven model-based PdM technique. Details on the implementation of each step in our case study are provided in Sect. 6.

Step 1: Data preprocessing. Raw diagnostic data are initially collected on a central server and preprocessed to improve their usability and relevance. Temporal alignment is performed to synchronize data points from various sensors or sources to a common timeline. Duplicate entries, which may arise from overlapping logs or redundant systems, are identified and filtered to prevent overrepresentation. A subset of diagnostic alarms is then selected based on their relevance to known failure modes, reducing noise and focusing on indicators with diagnostic significance.

Step 2: Cross-referencing diagnostic data with corrective maintenance logs. Historical corrective maintenance records are integrated with diagnostic data logs. This step enables the identification of correlations between specific alarms and actual failure events, providing a foundation for understanding failure patterns and causal relationships.

Step 3: Derivation of the fault tree structure. FTs model the hierarchical relationships between system components and potential failure modes. They can be derived from system schematics or safety analysis documents, ensuring alignment with system architecture and failure pathways.

Step 4: Association of alarms with failure events. Alarms from the diagnostic system are systematically linked to specific failure events. This step establishes a mapping that allows predictive algorithms to infer potential failures based on real-time alarm data.

Step 5: Extraction of time series. Time-series data capturing the temporal evolution of faults leading to failures are extracted from the combined logs. This dataset serves as a basis for analyzing the progression of faults over time and identifying early warning indicators.

Step 6: Estimation of probability distributions. The fault-to-failure time series are analyzed to compute time-to-fault and fault-to-failure PDFs, which quantify the likelihood of specific failures occurring within defined time intervals, providing insights into the system behavior under varying conditions.

Step 7: Evaluation using FaultFlow. The FT structure and the computed PDFs are used to define the SSFT of the system failure logic. Then, FaultFlow can be used to derive the duration CDF of failure processes (not necessarily top-level failures) and importance measures of faults.

4.1 Exploitability of Results

The time-to-failure CDFs (i.e., unreliability functions) derived by our analysis support maintenance planning and optimization in multiple ways. Through these applications, diagnostic data analyzed by our framework are effectively used to build a comprehensive PdM system capable of anticipating failures, optimizing maintenance schedules, and significantly enhancing overall system reliability while reducing operational costs and service disruptions.

Maintenance Planning. The reliability curves of TCUs can be mathematically combined to evaluate various redundancy policies, such as the train composition configurations discussed in Sect. 7.2. Specifically, by aggregating the coach-level unreliability functions according to specific redundancy rules (e.g., the constraint that non-functioning TCUs cannot be adjacent), maintenance engineers can quantitatively assess the reliability implications of different train configurations. This enables informed decisions about optimal fleet composition based on reliability requirements and operational constraints.

Threshold-Based Maintenance Triggering. A particularly valuable application of reliability functions is the establishment of threshold-based maintenance policies. By defining critical reliability thresholds (e.g., 0.9, 0.8, or 0.7), maintenance interventions can be automatically triggered when the computed system reliability falls below these levels. This approach transforms traditional time-based maintenance into a more efficient reliability-centered strategy.

Maintenance Resource Optimization. The reliability functions also facilitate optimal allocation of maintenance resources. By quantitatively predicting when specific components or subsystems will reach critical reliability thresholds, maintenance teams can prioritize interventions based on both criticality and timing, preventing the allocation of resources to components that still maintain acceptable reliability levels while ensuring timely attention to degrading subsystems.

4.2 Remarks

We notice that some of the steps 1–7 described above are tricky and require a deep expertise. At the same time, the quality of the initial data is also crucial. For example, while combining diagnostic and corrective maintenance data is a robust way to identify causal relationships between alarms and failures, this step assumes that corrective maintenance logs are detailed, accurate, and sufficiently granular to match with diagnostic data, which is not always the case. In fact, in our case study, corrective maintenance requests and logs are usually filled manually by train drivers and maintainers. Therefore, the effectiveness here depends heavily on the quality and completeness of the maintenance records.

The definition of the SSFT structure is also a crucial step. Using system schematics or safety analyses ensures that the SSFT structure is grounded in the system actual architecture. This step requires domain expertise and can be resource-intensive. The same problem occurs in associating alarms with failure events, which comprises a critical task for predictive models. This step assumes that historical data sufficiently captures the range of possible failure scenarios and that alarms are consistently reliable indicators to properly understand temporal dynamics and enabling time-based predictions. However, extracting these series requires well-labeled and time-stamped data, which can be difficult if the logs are incomplete or inconsistently recorded. It also assumes that the progression of faults can be meaningfully captured from the available data.

Finally, computing probability distributions of the duration of fault and failure processes is a standard means for quantifying uncertainty and making probabilistic predictions. However, the accuracy of these distributions depends on having a large and representative dataset modeling fault and failure behaviors. In this perspective, potential challenges include:

– **Data quality:** The entire process hinges on the availability of high-quality and well-structured data. On the other hand, inconsistent, sparse, or biased data can undermine the results.

- **Scalability:** Deriving the SSFT structure and manually associating alarms with failure events might not scale well for very large or complex systems.
- **Adaptability:** The process assumes that past patterns are indicative of future behavior. While generally valid, evolving system configurations or environmental conditions could reduce the accuracy of the predictions.
- **Expert involvement:** Several steps (e.g., SSFT structure derivation, alarm association) require domain expertise, which may introduce subjectivity or bottlenecks in the process.

5 Case Study Context

5.1 Railway Asset

Our case study is performed in collaboration with Trenord, a railway operator in Northern Italy, and involves a regional railway system comprising over 400 trains that serve approximately 700,000 passengers daily through approximately 2,200 service routes. The maintenance of this extensive rolling stock is conducted across six dedicated maintenance facilities.

Currently, maintenance needs are identified through a manual, expert-driven approach. The importance of PdM for this railway operator extends beyond organizational advantages in maintenance management and operational cost reduction; it represents a critical improvement in service reliability for passengers. The ability to predict failures would enable the operator to recall trains to maintenance facilities in advance, removing them from service before failures occur. In contrast, with the current reactive approach, both operators and passengers experience disruptions when failures happen during active service, necessitating corrective maintenance interventions that impact service continuity and passenger experience.

The trains we consider in this study, which make up most of the fleet, are equipped each with 18 diagnostic devices per vehicle, integrated with an onboard diagnostic platform responsible for capturing and transferring diagnostic data to a wayside system for analysis. While this system captures diagnostic information, it currently lacks an automated PdM mechanism capable of anticipating potential failures proactively.

This research focuses on developing a PdM solution specifically targeting the Traction Control Unit (TCU), a critical and complex on-board system.

5.2 Preliminary Data Analysis

Dataset Characteristics. The dataset provided by the railway operator encompasses two primary data types collected over a three-year period:

1. **Diagnostic data** (5,278,950 records):
   - Comprises fault, problem, and malfunction records for the TCU
   - Captures diagnostic events and alerts rather than precise sensor measurements

- Each data point is a tuple which includes:
  - Coach where the event occurred
  - Event timestamp
  - Alert type and code
  - Geospatial information (latitude, longitude)
  - Train velocity
  - Coach identification

2. **Maintenance data** (473 records):
   - Includes both corrective and scheduled maintenance activities
   - Corrective maintenance: Interventions resulting from service-related faults
   - Scheduled maintenance: Planned activities according to the rolling stock maintenance schedule

Initial Data Analysis. Our preliminary investigation focuses exclusively on onboard diagnostic data to assess the feasibility of developing a predictive model. The research currently prioritizes predicting the number of critical alerts, which domain experts consider an indicator of potential system failures.

6 Application of the Process to the Case Study

In this section, we illustrate the application of our methodology to the case study and we provide some lessons learned.

Step 1: Data preprocessing. Raw diagnostic data on trains are collected on a central server using multiple data acquisition boards on each train. These boards somehow overlap and can possibly detect the same event with slight time-disalignments, therefore the task of synchronizing data w.r.t. a common timeline can be tricky. An information that may be useful is the train position and the known train length, but for the purposes of our analysis we decided to apply a simpler algorithm. In particular, for each coach we extracted the data coming from different acquisition boards separately and we proceeded to data fusion only after cleaning and selection phases. A subset of diagnostic alarms is then selected based on their relevance to known failure modes, reducing noise and focusing on indicators with diagnostic significance. Here we focus on the alarms concerning the failure of main TCU subsystems. These alarms are shown to the driver on the Driver Machine Interface (DMI) and are the ones that may trigger a manual maintenance request. First, we filtered data by train, by coach, by causing system, and by data acquisition board.

Step 2: Cross-referencing diagnostic data with corrective maintenance logs. We considered logs containing the following information: data and author of the request; train and coach identifiers; system identifier and issue; and, closure date. These logs are filled out manually, sometimes with coarse information. Anyway, we used the maintenance logs to split raw data in temporal sequences going from the end of a corrective maintenance intervention

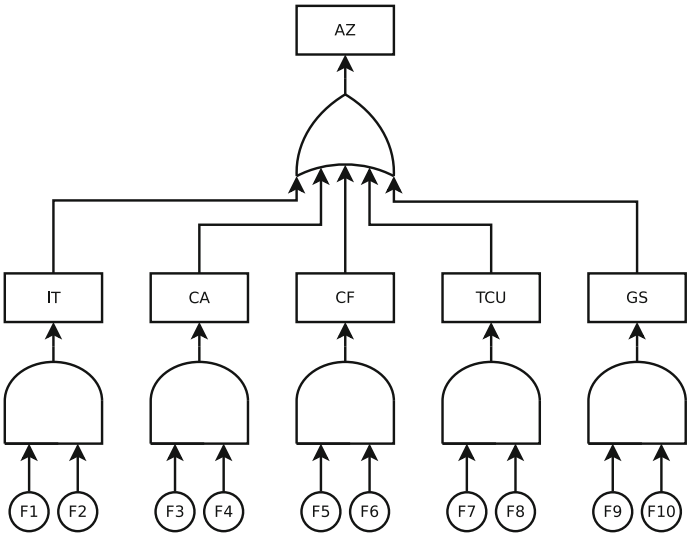

Fig. 2. High level system fault tree with main components.

to the raise of the first following request. This step helped us to identify correlations between specific alarms and to detect alarm sequences that lead to a system failure.

Step 3: Derivation of the fault tree structure. We constructed the SSFT structure on the basis of the list of alarms and electrical system schematics. Figure 2 shows the higher-level layer.

The top-level event is represented by the "AZ" box, which represents the system level failure or malfunction. The connections leading down from AZ represent the different potential causes or contributing factors. These are due to subsystem failures, in particular:

1. IT: Traction Inverter;
2. CA: Step down Chopper
3. CF: Braking Chopper
4. TCU: Train Control Unit board
5. GS: Inverter Static Group

Each of these lower-level subsystems (i.e., IT, CA, CF, TCU, GS) has further basic failure events represented by the numbered nodes F1, F2, ..., F10. The tree-like structure with AND gates models the fact that multiple lower-level failures would need to occur to ultimately lead to the top-level AZ failure.

In the traction system, no redundant components are inherently present. This might initially appear counterintuitive for a system with high availability requirements: indeed, redundancy is effectively implemented at the train composition level. More details are given in Sect. 7.2.

Step 4: Association of alarms with failure events. The railway operator provided a list with the identifier, description, and severity of each alarm. In

this study, we considered the high-severity alarms and linked them to specific subsystem failures. The alarms were partitioned into two groups, i.e., visible to the driver and visible only to the maintainers. The alarm description helped us to link the referred subsystem or component, but there were cases (especially among the lower-severity alarms) in which this connection was not clear. In this step, the help of the railway operator was crucial.

Step 5: Extraction of time series. In this step, the diagnostic data were partitioned into separate files, one for each coach. For every corrective maintenance record, we identified the first occurrence of each selected alarm (among the various data acquisition boards) and computed the time difference between the alarm and the preceding maintenance event. Specifically, let M_i and M_{i+1} be two consecutive maintenance events, with associated timestamps $M_i(t_i)$ and $M_{i+1}(t_{i+1})$. For each alarm type A_x, we selected the timestamp of its *first occurrence* within the interval $(M_i(t_i), M_{i+1}(t_{i+1}))$, denoted as $A_x(t_1)$, and computed the delay from the preceding maintenance:

$$\Delta t_x = A_x(t_1) - M_i(t_i)$$

For this case study, data from 23 coaches across 5 trains was extracted.

Step 6: Estimation of probability distributions. The inter-event times Δt_x were modeled using an exponential distribution. For each alarm type A_x, the goal is to estimate the parameter λ_x of the exponential distribution:

$$f(t; \lambda_x) = \lambda_x e^{-\lambda_x t}, \quad t \geq 0$$

The maximum likelihood estimate (MLE) of λ_x under complete (non-censored) data is:

$$\hat{\lambda}_x = \frac{n}{\sum_{i=1}^{n} t_i}$$

where t_i is the observed time to the first alarm after the i-th maintenance and n is the number of observation windows (M_i, M_{i+1}) where the alarm appeared.

However, not every alarm A_x occurs between every pair of maintenance events. In such cases, the alarm is said to be *right-censored*: we only know that the event did not occur within the observation window (M_i, M_{i+1}), but not whether or when it might occur after.

To correctly account for this censoring, we define:

- t_i: the time between M_i and the first alarm occurrence, or the duration of the observation window (if censored).
- $\delta_i \in \{0, 1\}$: an indicator variable, where $\delta_i = 1$ if the alarm occurred (i.e., uncensored), and $\delta_i = 0$ otherwise.

The likelihood function accounting for censoring becomes:

$$L(\lambda_x) = \prod_{i=1}^{n} \left[\lambda_x e^{-\lambda_x t_i}\right]^{\delta_i} \cdot \left[e^{-\lambda_x t_i}\right]^{1-\delta_i} = \lambda_x^{\sum \delta_i} e^{-\lambda_x \sum t_i}$$

Maximizing this likelihood yields the censored MLE estimate:

$$\hat{\lambda}_x = \frac{\sum_{i=1}^{n} \delta_i}{\sum_{i=1}^{n} t_i}$$

This approach correctly incorporates both observed and censored times, resulting in a statistically safe estimation of the distribution of the considered time parameter (i.e., time-to-fault or fault-to-failure time).

Step 7: Evaluation using FaultFlow. We implemented the model in the FaultFlow library to compute the time-to-failure CDF for the full AZ system and for the subsystems IT, CA, CF, TCU board, GS, as well as the Birnbaum and the Fussell-Vesely importance measures of faults. We also used the `librbd` library to perform reliability analysis.

7 Results and Discussion

We present the experimental results obtained by using the FaultFlow library and the `librbd` library to analyze our case study. All experiments were performed on an Apple M3 CPU 8 Core @ 4.06 GHz with 16 GB RAM, running MacOS. The experiments performed using the FaultFlow library took nearly 2.7 s, while those performed using the `librbd` library took nearly 15 ms.

7.1 TCU Unreliability Analysis

In the following, we present the results of the FaultFlow analysis for the considered TCU, including the system-level time-to-failure CDF (also referred to as the unreliability function) and the importance measures of component faults.

Figure 3 shows the time-to-failure CDF of the coach traction system, representing the probability that at least one alarm occurs within time t. The steep rise in the curve indicates a rapid decrease in reliability. Notably, after approximately 300 h, the probability of experiencing at least one critical alarm reaches 0.7. This signifies a significant degradation in reliability within the initial operational period. Furthermore, the curve approaches 1.0 around 750 h, suggesting that it is almost certain that at least one critical alarm occurs within this time interval. This early and substantial unreliability points out the need for proactive rather than reactive maintenance. Analyzing more details in the diagnostic logs in conjunction with this CDF enables the identification of specific alarm patterns and their frequency within the first 500 h of operation.

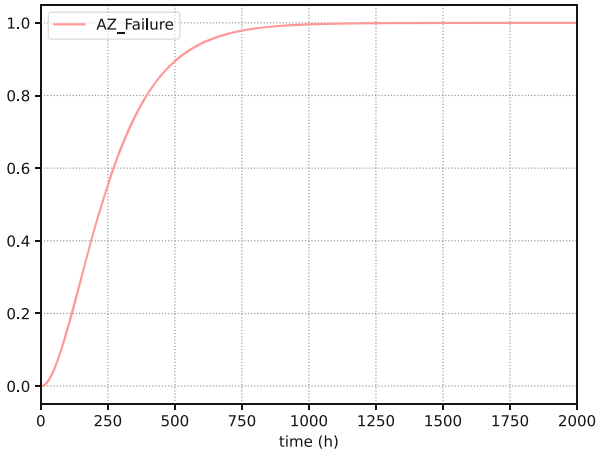

Fig. 3. CDF of the time to the AZ failure.

Figure 4 displays the Birnbaum importance measure for each fault, illustrating the relative influence of individual components on the system failure probability. It is evident that not all alarms contribute equally to the overall unreliability. Faults F3 and F4, associated with the step-down chopper, and F1 and F2, associated with the traction inverter, emerge as the most impactful. Specifically, F3 exhibits the highest Birnbaum importance measure, peaking around 10^6 h, followed closely by F4. This highlights their critical role in determining the system's overall reliability. Prioritizing the monitoring and maintenance of these components is crucial, as their failure has a more significant impact on the system's unreliability compared to other faults.

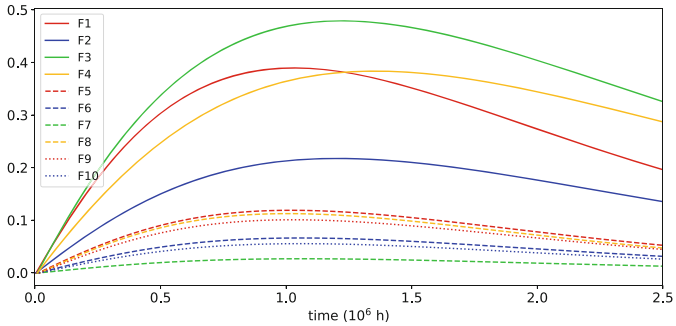

Fig. 4. Birnbaum importance measures of faults.

Figure 5 presents the Fussell-Vesely importance measure of each fault. Consistently with the Birnbaum measure, faults F3 and F4 dominate the ranking,

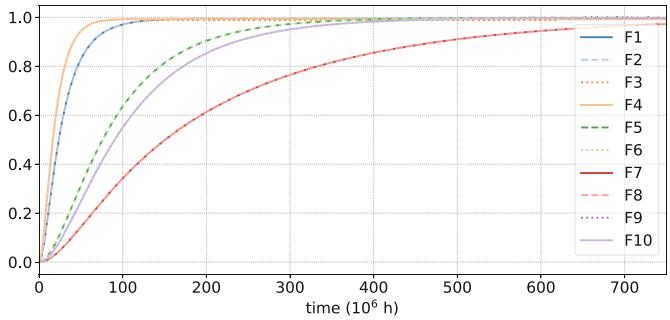

Fig. 5. Fussell Vesely importance measures of faults.

particularly within the early operational period. F3 and F4 show a rapid increase in their Fussell-Vesely importance measure, being close to 1.0 within 200–300 $\cdot 10^6$ h. This highlights their critical role in the failure behavior of the traction system and suggests that their failure is highly likely to lead to a system failure. This consistency across both importance measures strongly reinforces the need to prioritize these components in diagnostics and maintenance planning. This could involve more frequent inspections or sensor-based monitoring of critical components. The consistently high importance measures for F1, F2, F3, and F4 (Figs. 4 and 5 indicate that predictive maintenance efforts should be heavily concentrated on the traction inverter and step-down chopper).

7.2 Train Reliability Analysis

The considered trains are composed by powered coaches, each with a TCU, as shown in Fig. 6. Operational continuity can be maintained even with up to two TCUs excluded from service (except for the train configuration with three coaches, for which only one failed TCU is tolerated) provided that these non-functioning units are not positioned consecutively within the train configuration.

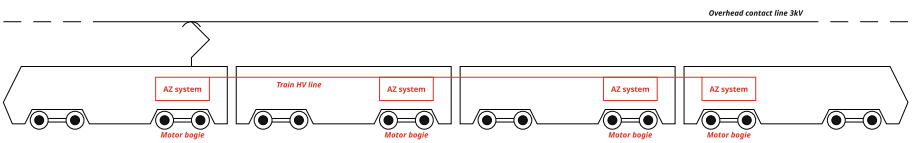

Fig. 6. A train configuration with 4 coaches.

We considered trains composed of 3, 4, 5 or 6 coaches. We used FaultFlow to derive the unreliability of the TCU of each coach, as described in Sect. 7.1, and then we used `librbd` to evaluate the overall train reliability. Specifically, the reliability of the 3-coach configuration is evaluated as the reliability of an RBD

consisting of a 2oo3 block (modeling the fact that 2 out of 3 TCUs are functioning); the reliability of the 4-coach configuration is evaluated as the reliability of the RBD shown in Fig. 7, consisting of the parallel composition of: a 3oo4 block (modeling the fact that 3 out of 4 TCUs are functioning), and three sequences of four blocks, modeling the conditions under which the system is functioning although 2 TCUs are non-functioning (i.e., the TCUs of coaches 1 and 3, or the TCUs of coaches 2 and 3 are functioning, or the TCUs of coaches 2 and 4 are functioning); and so on. Note that, in doing so, we are safely underestimating the system reliability, given that the same events (i.e., the fact that the TCU of a specific coach is not functioning) affect different compositions of blocks of the RBD, and that these events are dependent and positively correlated but are considered independent of each other in the quantitative evaluation [2].

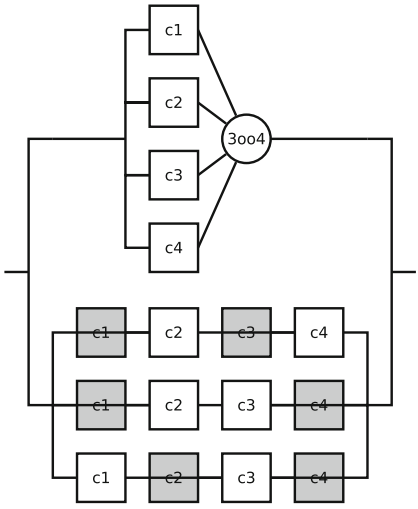

Fig. 7. RBD for a train with 4 coaches (grey blocks identify the failed coaches).

Figure 8 shows the reliability of the train configurations made of n coaches with $n \in \{3, 4, 5, 6\}$. A train with 3 coaches is the most reliable, due to the fact that with this configuration we have fewer combinations of possible failures and only the failure of a single coach is tolerated. For the configurations with 4, 5, and 6 coaches, the adjacency constraint becomes increasingly restrictive and significantly reduces the number of functioning system states. The adjacency requirement eliminates many potential operational configurations where non-adjacent coaches could theoretically maintain system functionality, thereby creating a more conservative reliability assessment. This fact also explains why the reliability curves for a larger number of coaches show steeper degradation patterns, as the system becomes more vulnerable to failures that violate the spatial continuity requirement rather than just the minimum operational capacity. Also note that the adjacency constraint serves as a simplification that may

be overly restrictive compared to usual operational flexibility, also making the obtained results an understimation of the actual reliability curves.

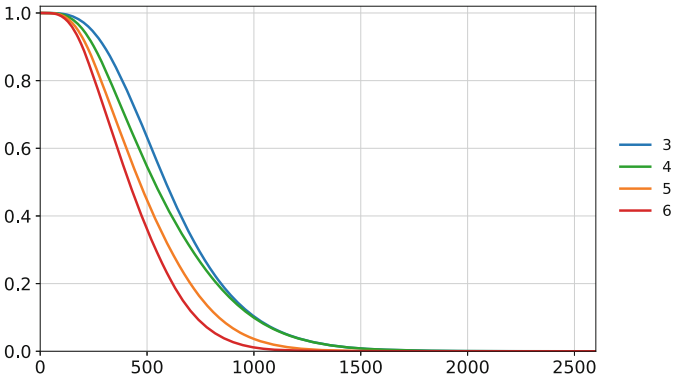

Fig. 8. Reliability of trains composed of 3, 4, 5 and 6 coaches.

8 Conclusions

We have presented a novel approach to derive stochastic fault tree models by strategically integrating extensive diagnostic data logs, maintenance records, and domain-specific knowledge. Then, we used these models to predict component and system-level reliability dynamics, supporting the definition of proactive maintenance strategies by exploiting existing diagnostic infrastructure, without requiring additional sensor investments. The feasibility and effectiveness of the approach are preliminarily demonstrated on a railway case study concerning the traction control unit of a fleet of regional passenger trains,

We have remarked how the success of the approach depends on the quality of data, discussing potential threats to validity. Specifically, addressing potential data limitations as well as validating the system extensively in real-world scenarios will be crucial for thoroughly assessing its reliability and effectiveness.

The study presented in this paper has been prompted by the need of railway industry for availability of efficient and robust techniques and tools for predictive maintenance. Our research presents a systematic approach to bridge the gap between diagnostic and predictive maintenance in railway vehicles. By developing a methodology that leverages existing sensor diagnostic data, we have demonstrated the potential of transforming legacy systems into proactive maintenance frameworks. In particular, key scientific contributions of our work include:

– establishing a replicable process for converting diagnostic data into predictive maintenance models;

– validating the feasibility of fault tree analysis for real-time failure prediction;
– showing the approach application in a case study using real-world data.

While our initial results are promising, several challenges and limitations warrant further investigation. In particular, this study leverages only a subset of available diagnostics alarms related to the TCU, hence this approach can be extended to other onboard systems to improve the accuracy of the overall reliability estimation. Future research directions include:

– exploiting machine learning algorithms that can dynamically adjust predictive models based on evolving system characteristics;
– investigating methods to improve data quality and reduce uncertainty in maintenance logs;
– exploring cross-domain applications of our predictive maintenance approach.

Acknowledgements. This study was carried out within the MUR PRIN 2022 PNRR P2022A492B project ADVENTURE (ADVancEd iNtegraTed evalUation of Railway systEms) and the MOST – Sustainable Mobility National Research Center and received funding from the European Union NextGenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4.

References

1. Arnold, F., Belinfante, A., Van der Berg, F., Guck, D., Stoelinga, M.: DFTCalc: a tool for efficient fault tree analysis. In: Bitsch, F., Guiochet, J., Kaâniche, M. (eds.) SAFECOMP 2013. LNCS, vol. 8153, pp. 293–301. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40793-2_27
2. Baccelli, F., Makowski, A.M.: Multidimensional stochastic ordering and associated random variables. Oper. Res. **37**(3), 478–487 (1989)
3. Biagi, M., Vicario, E., German, R.: Extending the steady state analysis of hierarchical semi-Markov processes with parallel regions. In: European Workshop on Performance Engineering, pp. 62–77 (2018)
4. Binder, M., Mezhuyev, V., Tschandl, M.: Predictive maintenance for railway domain: a systematic literature review. IEEE Eng. Manage. Rev. **51**(2), 120–140 (2023)
5. Boudali, H., Nijmeijer, A., Nijmeijer, A., Stoelinga, M.I.A.: DFTSim: a simulation tool for extended dynamic fault trees. In: 42nd Annual Simulation Symposium (ANSS 2009), p. 31. Association for Computing Machinery (2009)
6. Budde, C.E., Ruijters, E., Stoelinga, M.: The dynamic fault tree rare event simulator. In: Gribaudo, M., Jansen, D.N., Remke, A. (eds.) QEST 2020. LNCS, vol. 12289, pp. 233–238. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59854-9_17
7. Carnevali, L., Cerboni, S., Montecchi, L., Vicario, E.: Faultflow: an MDE library for dependability evaluation of component-based systems. IEEE Trans. Dependable Secure Comput. 1–18 (2025). https://doi.org/10.1109/TDSC.2025.3532340
8. Carnevali, L., Cerboni, S., Picano, B., Scommegna, L., Vicario, E.: An observation metamodel for dependability tools. In: 2024 19th European Dependable Computing Conference (EDCC), pp. 169–172. IEEE (2024)

9. Carnevali, L., Ciani, L., Fantechi, A., Gori, G., Papini, M.: An efficient library for reliability block diagram evaluation. Appl. Sci. **11**(9) (2021). https://doi.org/10.3390/app11094026
10. Carnevali, L., German, R., Santoni, F., Vicario, E.: Compositional analysis of hierarchical UML statecharts. IEEE Trans. Soft. Eng. **48**(12), 4762–4788 (2021)
11. Carnevali, L., Giandomenico, F.D., Fantechi, A., Gnesi, S., Gori, G.: Quantitative dependability evaluation of train control systems in presence of uncertainty: a systematic literature review. IEEE Trans. Intell. Transp. Syst. **26**(4), 4298–4314 (2025). https://doi.org/10.1109/TITS.2025.3530112
12. Chen, H., Jiang, B.: A review of fault detection and diagnosis for the traction system in high-speed trains. IEEE Trans. Intell. Transp. Syst. **21**(2), 450–465 (2020). https://doi.org/10.1109/TITS.2019.2897583
13. Chiacchio, F., Aizpurua, J.I., Compagno, L., D'Urso, D.: Shyftoo, an object-oriented Monte Carlo simulation library for the modeling of stochastic hybrid fault tree automaton. Expert Syst. Appl. **146**, 113139 (2020)
14. Davari, N., Veloso, B., Ribeiro, R.P., Pereira, P.M., Gama, J.: Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–10 (2021). https://doi.org/10.1109/DSAA53316.2021.9564181
15. Deavours, D.D., et al.: The Mobius framework and its implementation. IEEE Tran. Soft. Eng. **28**(10), 956–969 (2002)
16. Ferdous, R., Spagnolo, G., Borselli, A., Rota, L., Ferrari, A.: Identifying maintenance needs with machine learning: a case study in railways. In: 2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW), pp. 22–25 (2024). https://doi.org/10.1109/REW61692.2024.00008
17. Homm, D., German, R.: Analysis of hierarchical semi-Markov processes with parallel regions. In: Remke, A., Haverkort, B.R. (eds.) MMB&DFT 2016. LNCS, vol. 9629, pp. 92–106. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31559-1_9
18. Kabir, S.: An overview of fault tree analysis and its application in model based dependability analysis. Expert Syst. Appl. **77**, 114–135 (2017)
19. Librbd Library (2025). https://github.com/marcopapini/librbd
20. Liu, J., Zhang, Y., Han, J., He, J., Sun, J., Zhou, T.: Intelligent hazard-risk prediction model for train control systems. IEEE Trans. Intell. Transp. Syst. **21**(11), 4693–4704 (2020). https://doi.org/10.1109/TITS.2019.2945333
21. Manno, G., Chiacchio, F., Compagno, L., D'Urso, D., Trapani, N.: MatCarloRe: an integrated FT and Monte Carlo Simulink tool for the reliability assessment of dynamic fault tree. Expert Syst. Appl. **39**(12), 10334–10342 (2012)
22. Manno, G., Chiacchio, F., Compagno, L., D'Urso, D., Trapani, N.: Conception of repairable dynamic fault trees and resolution by the use of RAATSS, a Matlab® toolbox based on the ATS formalism. Reliab. Eng. Syst. Saf. **121**, 250–262 (2014)
23. Montani, S., Portinale, L., Bobbio, A., Codetta-Raiteri, D.: RADYBAN: A tool for reliability analysis of dynamic fault trees through conversion into dynamic Bayesian networks. Reliab. Eng. Syst. Saf. **93**(7), 922–932 (2008)
24. Nauta, M., Bucur, D., Stoelinga, M.: LIFT: learning fault trees from observational data. In: McIver, A., Horvath, A. (eds.) QEST 2018. LNCS, vol. 11024, pp. 306–322. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99154-2_19
25. Paolieri, M., Biagi, M., Carnevali, L., Vicario, E.: The ORIS tool: quantitative evaluation of non-Markovian systems. IEEE Trans. Softw. Eng. **47**(6), 1211–1225 (2021)

26. Parri, J., Sampietro, S., Vicario, E.: FaultFlow: a tool supporting an MDE approach for timed failure logic analysis. In: European Dependable Computing Conference, pp. 25–32. IEEE (2021)
27. Pyramis Library (2025). https://github.com/oris-tool/pyramis
28. Ribeiro, R.P., Pereira, P., Gama, J.: Sequential anomalies: a study in the railway industry. Mach. Learn. **105**, 127–153 (2016)
29. Ruijters, E., Stoelinga, M.: Fault tree analysis: a survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. **15**, 29–62 (2015)
30. Salfner, F., Lenk, M., Malek, M.: A survey of online failure prediction methods. ACM Comput. Surv. **42**(3), 1–42 (2010)
31. Stamatis, D.H.: Failure Mode and Effect Analysis: FMEA from Theory to Execution. Quality Press (2003)
32. Trivedi, K.S., Sahner, R.: Sharpe at the age of twenty two. ACM SIGMETRICS Perform. Eval. Rev. **36**(4), 52–57 (2009)
33. Verbeek, E., Fahland, D.: CPN IDE: an extensible replacement for CPN Tools that uses Access/CPN. In: International Conference on Process Mining Doctoral Consortium and Demo Track, pp. 29–30 (2021)
34. Volk, M., Sher, F., Katoen, J.P., Stoelinga, M.: SAFEST: fault tree analysis via probabilistic model checking. In: Annual Reliability and Maintainability Symposium (RAMS), pp. 1–7. IEEE (2024)
35. Wang, Y., Du, X., Lu, Z., Duan, Q., Wu, J.: Improved LSTM-based time-series anomaly detection in rail transit operation environments. IEEE Trans. Industr. Inf. **18**(12), 9027–9036 (2022)
36. Zimmermann, A.: Modelling and performance evaluation with TimeNET 4.4. In: Bertrand, N., Bortolussi, L. (eds.) QEST 2017. LNCS, vol. 10503, pp. 300–303. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66335-7_19