International Journal on Software Tools for Technology Transfer (2022) 24:331–350
https://doi.org/10.1007/s10009-022-00652-4

GENERAL

Special Issue: FMICS 2019/2020

DFT modeling approach for operational risk assessment of railway infrastructure

Norman Weik[1] · Matthias Volk[2] · Joost-Pieter Katoen[3] · Nils Nießen[4]

Accepted: 8 March 2022 / Published online: 5 April 2022

Abstract

Reliability engineering of railway infrastructure aims to understand failure processes and to improve the efficiency and effectiveness of investments and maintenance planning such that a high quality of service is achieved. While formal methods are widely used to verify the design specifications of safety-critical components in train control, quantitative methods to analyze the service reliability associated with specific system designs are only starting to emerge. In this paper, we strive to advance the use of formal fault-tree modeling for providing a quantitative assessment of the railway infrastructure's service reliability in the design phase. While, individually, most subsystems required for route-setting and train control are well understood, the system's reliability to globally provide its designated service capacity is less studied. To this end, we present a framework based on dynamic fault trees that allows to analyze train routability based on train paths projected in the interlocking system. We particularly focus on the dependency of train paths on track-based assets such as switches and crossings, which are particularly prone to failures due to their being subject to weather and heavy wear. By using probabilistic model checking to analyze and verify the reliability of feasible route sets for scheduled train lines, performance metrics for reliability analysis of the system as a whole as well as criticality analysis of individual (sub-)components become available. The approach, which has been previously discussed in our paper at FMICS 2019, is further refined, and additional algorithmic approaches, analysis settings and application scenarios in infrastructure and maintenance planning are discussed.

Keywords Dynamic fault trees · Railways · Risk assessment · Railway infrastructure · Routability

1 Introduction

Highly reliable infrastructure components are a fundamental requirement for punctual high-quality railway operations. Given long renewal cycles and intricate planning procedures in creating new or adjusting existing infrastructure, strategic decisions in the design of track layouts and signaling systems are extremely critical. At the same time, wayside assets for track occupation detection or turnout system are subject to environmental effects. As a result, failures cannot be totally avoided and should be accounted for in infrastructure and maintenance planning to ensure the robustness and resilience of train services.

For many years, formal verification of both architecture and components of signaling systems has successfully been applied by manufacturers to satisfy industry standards and

Supported by German Research Foundation (DFG) with Research Training Group 2236 "UnRAVeL" and partially supported by NWO under the grant "PrimaVera" number NWA.1160.18.238.

✉ Norman Weik
norman.weik@dlr.de

Matthias Volk
m.volk@utwente.nl

Joost-Pieter Katoen
katoen@cs.rwth-aachen.de

Nils Nießen
niessen@via.rwth-aachen.de

[1] Institute of Transportation Systems, German Aerospace Center (DLR), Braunschweig, Germany

[2] Formal Methods and Tools (FMT), University of Twente, Enschede, The Netherlands

[3] Chair of Software Modeling and Verification, RWTH Aachen University, Aachen, Germany

[4] Institute of Transport Science, RWTH Aachen University, Aachen, Germany

safety-integrity level (SIL) requirements for safety-critical railway applications [4,12]. The use of formal methods for predictive assessment of service-related (non-critical) effects of infrastructure component failures is less common. Assessment based on expert opinions and heuristic approaches, e.g., referring to the number of train runs or the availability of alternative routes, continues to be widespread in this area (see, e.g., [20]). A formal, quantitative a-priori assessment of infrastructure reliability and the effects of failures on train operations, in the design phase, is missing, even though the need for formalization of RAMS (reliability, availability, maintainability, safety) analysis has been formulated in industry standards including rail-specific CENELEC norms EN 50126 [15] and EN 50129 [14].

We present an approach based on *dynamic fault trees (DFTs)* [53], a dynamic extension of classical (static) fault trees. The approach quantitatively analyzes the reliability of train operations based on the availability of feasible train paths projected in the interlocking system for given train lines. In order to operate train paths, wayside infrastructure components, so-called *field elements*, need to be in an operable route-conforming state. Our approach, while not limited to this area, focuses on these components including switches, signals, or train detection systems, that are subject to weather and wear and cause a significant share of service-affecting failures [6].

The approach is suitable to pinpoint critical components that should receive particular emphasis in condition monitoring and maintenance planning. *Criticality* in this context refers to the *service aspect*, *safety* is assumed to be granted by the signaling system. As a result of our analysis, different track layouts, scheduled line plans or asset management decisions can be compared based on the underlying train routes that determine train operations. In particular, weaknesses in railway track layouts and routing concepts can be identified and it can be analyzed how changes of component reliability transform into the reliability of system as a whole.

An overview of the approach is provided in Fig. 1. From given railway infrastructure data, possible routes for each train type and the associated way-side infrastructure components are extracted. Train routes are then mapped to and modeled by a DFT. For each type of field element, individual DFTs modeling the corresponding *component* failure behavior are provided. All DFTs are automatically combined into one complete DFT for the entire station area by connecting routes with the corresponding field elements required for successful operation. The focus of our work is on local, wayside infrastructure elements required for train routing and control. Systems such as train communications or energy supply are not considered, but could theoretically be incorporated.

The complete DFT can be analyzed with off-the-shelf DFT analysis tools, such as Storm[59] or DFTRES[11]. In our approach, we use the framework of Storm, which performs probabilistic model checking [2,40]. The DFT is first simplified by rewriting the graph structure [34]. Afterward, Storm generates a continuous-time Markov chain (CTMC) that captures the behavior of the DFT. The CTMC is analyzed with respect to a given metric. Supported metrics include the unreliability of the station area, the mean-time-to-failure (MTTF), or the criticality of single field elements. The analysis within Storm uses efficient, state-of-the-art model checking techniques for probabilistic models [31]. In case the fault tree only contains static behavior, we can use binary-decision diagrams (BDDs) to efficiently analyze the (static) fault tree [50]. The computed analysis results offer insights into the reliability of the station area and the criticality of field elements with respect to the routing possibilities. These findings can be used in tactical or strategic planning of infrastructure design and asset management.

This paper is an extended version of the conference paper [60]. The major extensions in this paper are:

1. An improved version of the pipeline illustrated in Fig. 1 featuring enhanced algorithmic procedures. In particular, static-fault-tree analysis using binary decision diagrams is included in the analysis framework. The fault trees are static when failure processes are aggregated to the field-element level and no sequence dependencies are contained in the model. In this context, the respective numerical performance and assessment quality of the corresponding models and solution techniques are investigated.
2. New reliability assessment metrics and analysis settings. In particular, the usability and predictive quality of the reliability assessment and the effect of different reliability time horizons are investigated.
3. A more detailed discussion of the underlying premises, methodological limitations, and practical implications of the criticality assessment framework. To this end, new application scenarios in different variants are introduced and discussed.

The paper is arranged as follows: We start by reviewing the previous use of formal methods in railway infrastructure planning as well as the current status of standards and methodology for reliability analysis of railway systems in Sect. 2. In Sect. 3, we give a brief overview on basic notions of dynamic fault trees, the underlying stochastic model and introduce the analysis of DFTs via probabilistic model checking. Subsequently, the high-level DFT for the routing options in a railway station as well as the DFTs for the individual field elements are discussed in Sect. 4. The considered quality metrics for the analysis are specified in Sect. 5. We evaluate our approach on four German railway stations according to different reliability metrics and model setups. We introduce the settings in Sect. 6 and present and discuss the results

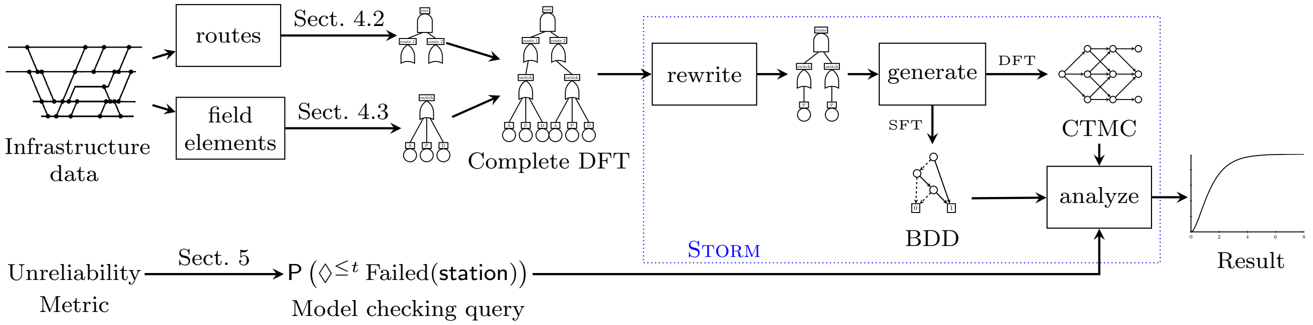


Fig. 1 Overview of the DFT-based analysis approach for railway stations

in Sect. 7. We discuss the general methodology in Sect. 8. The paper concludes with a discussion of the model's current scope and an outlook towards possible future extensions.

2 Literature review

Railway operations in densely operated networks are effectuated by signaling systems, which are based on an interlocking system as a fundamental building block. The interlocking system ensures the proper setting of switch positions, compatibility of train routes and safety of operations and prohibits the controller from setting conflicting routes [24]. Technically, interlocking and signaling systems rely on a combination of subsystems required for train positioning, verifying train integrity, setting the route and communicating the movement authority to the train. As a consequence, a series of preconditions has to be met before a route can be set for a train. As the systems are safety critical, formal methods are widely used for hardware and software verification in this area (see [56] for a systematic overview).

2.1 Verifying signaling systems

An important aspect of railway signaling system analysis is the verification of interlocking control tables specifying the dependencies and restrictions in setting train paths. In [24], interlocking control information represented in ladder logic diagrams by Westinghouse Signal was transformed to Boolean expressions that could be verified using theorem provers. James and Roggenbach [33] and Kanso et al. [38] used SAT model checking for the same systems. Colored Petri nets were used in [57] to model railway systems based on a two-level approach. It comprises (a) the signaling layout describing the physical infrastructure as well as train movements and (b) the interlocking featuring the functional dependencies between elements as described by control tables. Abstract state machines in combination with NuSMV and symbolic model checking have been used in [28] and [63], adopting a high-level view focusing on train operations in stations. Ferrari et al. [22] explored the limits of applicability of different model-checking tools for railway-interlocking control-table verification. A SIL-based suitability review of formal methods and tools for railway applications has been given in [21].

With the emergence of modern radio-based train control systems including ETCS (European Train Control Systems), formal approaches for verifying system properties have seen another rise. Whereas previously, the focus had been on the representation of infrastructure-component dependencies, train-radio communications and vehicle-based systems for automated train control (ATC) have received new attention.

In [47], a controller for the ETCS cooperation protocol is defined and controllability, safety, liveness and reactivity of the system are analyzed using deductive verification. Cimatti et al. [17] assess the consistency of ETCS specifications and requirements as a hybrid system incorporating train movements by turning to temporal logic in combination with regular expressions. A statechart modeling approach is presented in [32]. Discrete event simulation is used to perform reliability analysis w.r.t. quality standards for radio transmission. Biagi et al. [8] perform communication failure modeling on the highest evolution of fully GPS/radio-controlled train operation in ETCS level 3 based on stochastic Petri nets. In [5], a similar ETCS Level 3 setting with moving block signaling is considered using UPPAAL SMC for the analysis. In a comparative compilation, 9 different formal tools are assessed and compared with respect to their applicability for verifying modern moving block signaling systems [23].

Standardization of railway signaling design and verification requirements is given in CENELEC norms. In EN 50128 [13], even at SIL 1 and 2, formal methods for software applications are recommended. The same holds true for EN 50126-1 [15] defining rules and guidelines for risk and asset management of railway systems.

2.2 Risk and reliability assessment of railway systems

Risk and criticality assessment for railway assets currently is predominantly performed using structured approaches such as Failure Mode and Effect Analysis (FMEA). Hassankiadeh [29] and Kassa [39] develop corresponding schemes for track and switch failure modes. Prescott and Andrews [48] discuss Markov models for modeling the degradation of rails and rail foundations. In [1], an extended Petri-net framework is presented for this task.

Morant et al. [44] present a Markov model for switching between operational, degraded, and non-operational states for risk mitigation. A simulation approach for risk and availability assessment of railway infrastructure is discussed in [55]. A CTMC-modeling technique for joint reliability and performance analysis of railway stations and networks locally has been discussed in [61,62]. Khaled et al. [42] present an optimization framework for criticality assessment of links, hubs, and stations in American freight networks, where criticality is measured by a delay-based cost function in a mixed-integer-programming (MIP) setting. Similar data-driven approaches to criticality and resilience analysis of rail networks are frequently found in the literature (see [7] for an overview).

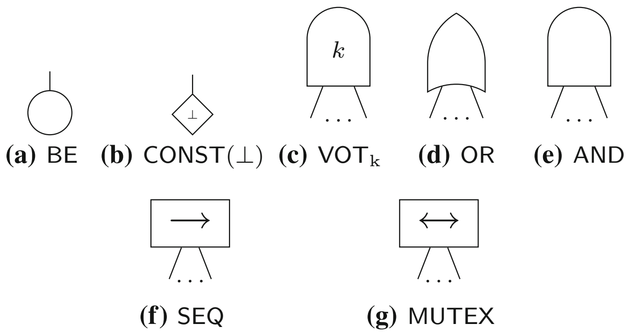
Fault trees are discussed in EN 50126 as an alternative for risk and reliability modeling of railway systems. Henry [30] uses this class of models to analyze train protection systems in metro systems. Chen et al. [16] use (static) fault trees to model the rail power supply system. Binary decision diagrams are used to solve the model.

More recently, [26] discusses an extended DFT model with dynamic gates allowing to model functional dependencies and replacement parts. Using probabilistic model checking the approach is applied in an analysis of general railway failures and corresponding maintenance strategies. A further extension focusing on a more detailed assessment of maintenance strategies is discussed in [52]. Here, the focus is on insulating joints in railway tracks which are important in delimiting track current circuit segments used for track occupation detection.

3 Principles of DFT modeling

3.1 Fault trees

Fault trees [53,54] (FTs) are directed acyclic graphs (DAG) with typed nodes (AND, OR, etc.). We call nodes of type T "a T". The successors of a node v in the DAG are called the *children* of v. Nodes without children are *basic events* (BEs), nodes with children are *gates*. We say a BE "*fails*", if the basic event occurs. Such failures are governed by probability distributions. Similarly, a gate fails if the failure condition over the children holds. The *top-level event* (TLE(F)) of fault tree F is a specifically marked node. We write TLE if F is clear from context. We say the "FT F fails" iff TLE(F) fails.



(a) BE **(b)** CONST(⊥) **(c)** VOT$_k$ **(d)** OR **(e)** AND

(f) SEQ **(g)** MUTEX

Fig. 2 Node types in static (first row) and dynamic (all) FTs

We recapitulate the different node types in fault trees as presented in [25]. A graphical representation of the relevant node types is given in Fig. 2. We refer to [36] for a detailed description of the semantics of FTs.

3.1.1 Static fault trees

Static fault trees (SFTs) have node types BE and VOT$_k$.

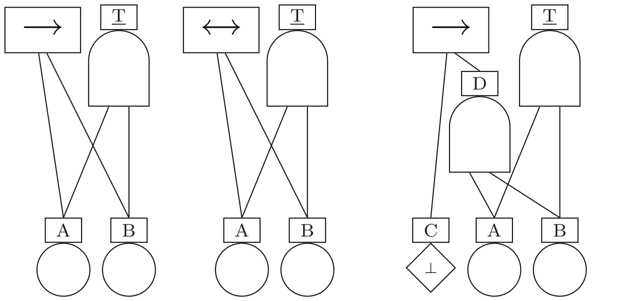
Basic events. BEs (Fig. 2a) represent atomic system components which fail according to an exponential failure distribution defined by the *failure rate*. *Constant fail-safe* BEs (CONST(⊥), Fig. 2b) are a special case of BEs that never fail.

Voting gates. The *voting gate* VOT$_k$ with threshold k (Fig. 2c) is the most general gate in SFTs. A VOT$_k$-gate fails, if at least k of its children have failed. Two special cases of the VOT$_k$ exist: the OR-gate (Fig. 2d) can be represented by a VOT$_1$-gate and fails if at least one child has failed. The AND-gate (Fig. 2e) with n children can be represented by a VOT$_n$-gate and fails if all its children have failed.

3.1.2 Dynamic fault trees

While SFTs are widely used in industry, they lack expressive power to faithfully model many aspects of complex systems such as spare management, order-dependent failures, functional dependencies or failure restrictions. *Dynamic fault trees* (DFTs) [18] are a commonly used extension of SFTs with several new node types supporting these aspects. In the following, we only introduce the DFT node types occurring in our models. A complete list of DFT node types is given in [36]. Note that the ordering of the children is relevant in DFTs, and we therefore order them from left to right.

Restrictors limit the possible failures of events. The *sequence enforcer* (SEQ, Fig. 2f) only allows failures of its



(a) SEQ **(b)** MUTEX **(c)** SEQ models MUTEX

Fig. 3 Examples of restrictors

children from left to right. For instance, in Fig. 3a, BE B is only allowed to fail if A has failed before.

The *mutual exclusion restriction* (MUTEX, Fig. 2g) is a special case of the SEQ. A MUTEX only allows one of its children to fail. An example is given in Fig. 3b. If A has failed, the MUTEX prevents the failure of B, and vice versa. MUTEX are syntactic sugar [35] and can be modeled with a SEQ and a fail-safe BE as shown in Fig. 3c. The first (fail-safe) child C of the SEQ-gate can never fail and therefore the failure of the second child D is always prevented. If for example A fails in the example, B cannot fail anymore as otherwise the failure of D would violate the failure order of the SEQ.

3.2 Markov chains

For the analysis, the DFTs are translated into *continuous-time Markov Chains (CTMCs)* [3].

Definition 1 (CTMC) A CTMC $\mathcal{C}$ is a three-tuple $\mathcal{C} = (S, R, L)$ with

- S a finite set of states,
- $R\colon S \times S \to \mathbb{R}_{\geq 0}$ the *transition rate matrix*, and
- $L : S \to 2^{AP}$ a labeling function assigning a set of *atomic propositions* $L(s) \subseteq AP$ to each state $s \in S$.

The *residence time* in a state s is defined by the negative exponential distribution parameterized by the *exit rate* $\sum_{s' \in S} R(s, s')$. *State labels* are associated to states and are used to identify specific states. For instance, the atomic proposition A_{fail} could be added to all states where DFT element A has failed.

Example 1 (CTMC) Figure 4 depicts an example CTMC with 5 states, transition rates and labels. The exit rate for state s_0 is $4+5+3 = 12$. State s_1 is labeled with $L(s_1) = \{A\}$, state s_4 is labeled with $L(s_4) = \{T\}$.

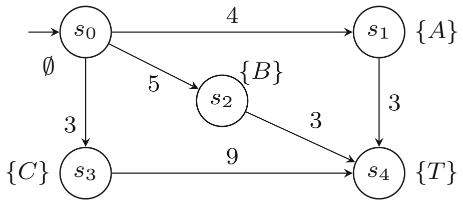


Fig. 4 Example CTMC

3.3 DFT analysis by model checking techniques

The general approach to analyzing DFTs is depicted in Fig. 1. Starting from a given DFT, the complete analysis follows three steps: (1) simplifying the DFT by rewriting, (2) generating the state space and (3) analyzing the resulting CTMC by model checking techniques. The three steps are implemented within the model checker Storm [31] and fully automated. We shortly introduce each of the analysis steps in the following. A more extensive description can be found in [41].

Simplifying DFTs. DFTs are usually created manually or semi-automatically with human guidance. As a result, the structure of DFTs is not optimized for analysis. As a preprocessing step, we simplify the structure of the DFT by employing the graph rewriting framework of [34]. Rewriting simplifies the graph structure by, e.g., removing superfluous levels of ANDs or ORs, or by merging multiple BEs into a single BE. These transformations preserve the semantics of the original DFT while improving the analysis performance.

State space generation. From the simplified DFT, the corresponding state space of a CTMC is generated. Several approaches have been presented for the state-space generation [10,18,59]. The approach implemented in Storm [59] exhaustively explores all possible sequences of failures. Starting from an initial state where all DFT elements are operational, new states are generated by letting one of the BEs fail and propagating the corresponding failure through the DFT. The transition rate between two states then corresponds to the failure rate of the newly failed BE. Labels are used to identify states where specific components or the TLE have failed.

Generating the state space is computationally the most expensive part as the state space can grow very large. Several optimization techniques are employed to mitigate the state-space explosion problem. Examples are exploiting symmetric structures [59], independently analyzing subtrees through modularization [27] and only exploring relevant failures [59]. It is also possible to only generate a partial state space by, e.g., restricting the number of considered consecutive failures [59]. Unexplored states then also correspond to a TLE failure.

Analysis via model checking. CTMCs are analyzed w.r.t. model checking queries that formalize the metrics we are interested in. The queries are specified in *continuous stochastic logic (CSL)* with reward extensions [3]. Given a CTMC and a set of model checking queries, the analysis is performed by applying standard algorithms from probabilistic model checking [3].

For specific types of fault trees, more efficient analysis techniques are possible. Static fault trees (SFTs), for instance, can be efficiently analyzed using binary decision diagrams (BDDs) [50,53]. The approach directly translates an SFT into a BDD—without the need to generate the state space first—and thus, scales better.

4 DFT model for reliability analysis of railway infrastructure

In the following, we present our DFT modeling technique for reliability analysis of railway infrastructure. The model builds on train routing and wayside assets required in train operations and control.

4.1 Modeling train runs

In railway operations, a centralized railway control system grants movement authority to trains, based on the verification of the safety of operations. This requires the availability and functionality of permanent way components, such as tracks, switches and crossings. In addition, field elements required for train movement detection and communications, including, e.g., signals, axle counters, and track circuits, are also required to be in a proper operational state, which is ensured by interlocking. In modern signaling systems like ETCS, parts of the control system are transferred to train-based on-board units which verify, e.g., train position. Still, the basic dependencies on the operability and correct state of field elements relevant for train routing remain valid.

In order to operate a train between two points, a valid route needs to be assigned to the train. This route may consist of one or several *train paths* associated with segments to which movement authority can be granted. In Fig. 5, this principle is illustrated for traditional signal-controlled train operations, where movement authority is communicated to the train driver by means of (main) signals delimiting coherent train paths. For a train to be safely operated, preconditions regarding the state of field elements have to be met. The failure processes of way-side field elements hence determine the availability and reliability of train paths, and consequently, train routing options.

4.2 DFT model for railway infrastructure analysis

Following the basic requirements for train routing, our fault tree model for infrastructure reliability analysis is based on train routing options for a given infrastructure. *Sets of viable routes* are established for each train type, e.g., restricting all topologically feasible routes to routes associated with platforms in case of passenger trains with a scheduled stop. Amongst suitable routes, an ordering based on train type-specific priorities is performed. In the context of this paper, priorities are supplied with the infrastructure data; in case corresponding data do not exist, priorities can, e.g., be derived based on route length, similarity to scheduled route or frequency of use.

Railway stations, which are among the most critical points in the railway network, require and provide the richest infrastructure. Each train route is subdivided into at most two train paths—one corresponding to a station entry and another corresponding to station exit. In case trains start or end in the station, only one train path is present. Further subdivision based on additional signals or smaller movement authorities may exist in practice and could also be considered in the model.

The routing options on the infrastructure are modeled by a fault tree. Figure 6 depicts an exemplary DFT model for a railway station.

The top-level event (depicted in red color) corresponds to a failure of the complete station area. We consider the station failed if at least one train type cannot be successfully routed anymore. This is modeled by an OR-gate over the possible route sets. Each route set in turn is considered failed if none of its routes is available anymore. This is modeled by an AND-gate. Each route consists of one or two train paths tp. A route fails if one or both of its train paths becomes unavailable. Note that train paths can be used in different routes. Failure of a train path can therefore render multiple routes unavailable at the same time.

A train path tp is considered failed if at least one of the required field elements along the path, e.g., switch, crossing, signal, has failed. For switches and slip switches, the track of the component corresponding to the train's route has to be unavailable. For instance, a train path going over the main track of a switch requires the DFT element switch main to be available. If only the branch track is unavailable, i.e., switch branch is failed, the train path is still available as the main track can still be used. Field elements can also occur in multiple train paths.

The presented DFT provides a high-level view on the routing options in a railway station area. By including the (physical) field elements, we model the influence of component failures on the routability of train types.

4.3 DFT models for infrastructure components

The field elements in a station area are modeled by DFTs as well. Our modular approach allows to create DFTs independently for each field element. Moreover, it is straightforward

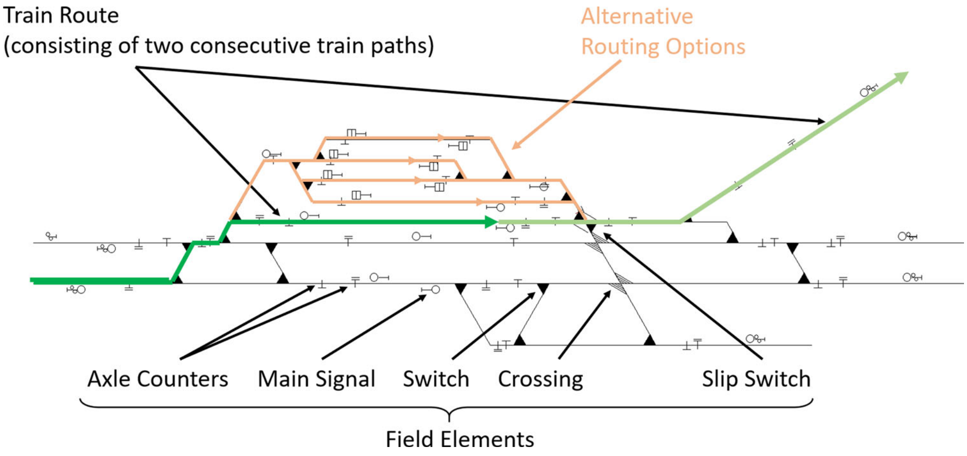


Fig. 5 Example of train routing in station areas. The overall route consists of two train paths (for entering/exiting the station area). A non-exhaustive selection of field elements associated with train control including switches, signals and axle counters for train detection is also depicted

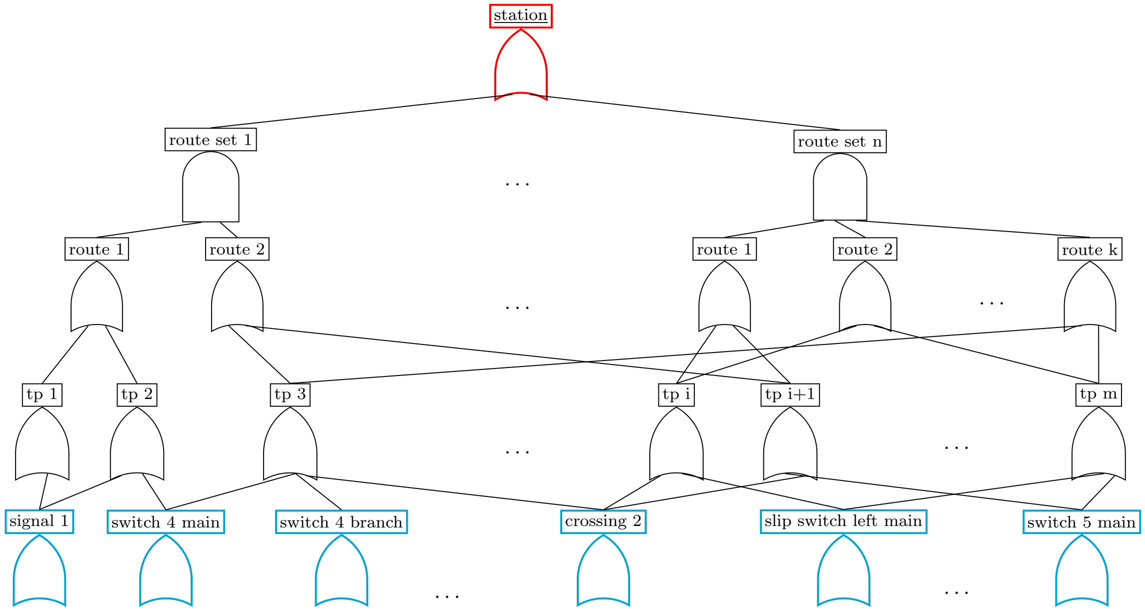


Fig. 6 Railway station fault tree

to include existing fault tree models—provided by the manufacturers for example.

In the following, we present DFT models for all relevant wayside infrastructure elements. We focus in particular on switches as they are most important for routing and therefore model them in greatest detail.

4.3.1 Switches

Switches connect multiple tracks such that trains can use either the *main track* or the *branching track*. The direction depends on the current state of the switch blades. The switch blades can be changed—and hence the routing—by moving them into the new position and locking them. The locking mechanism ensures that the blades cannot move underneath a train and therefore prevents derailment. Switches can be partially unavailable, i.e., one direction cannot be used anymore while the other one is still available. This can happen, for instance, if the switch engine cannot move the blades anymore to the other position but can still be moved back and safely locked in their current position. In that case, the current route can still be used.

We depict the DFT for a switch in Fig. 7. We distinguish between the failure of the main track (switch main) and the branch track (switch branch). Failure of the complete switch can be modeled by adding an OR-gate with switch main and switch branch as children. Both switch directions are connected to the train paths that use them. Both elements can occur as children of multiple train paths.

Each switch track can fail due to two reasons: (1) either the switch is *stuck* in the other direction (branch stuck/main stuck) and cannot switch anymore, or (2) a global failure occurs (global fail), which simultaneously renders both directions unavailable. We use the MUTEX to ensure that the

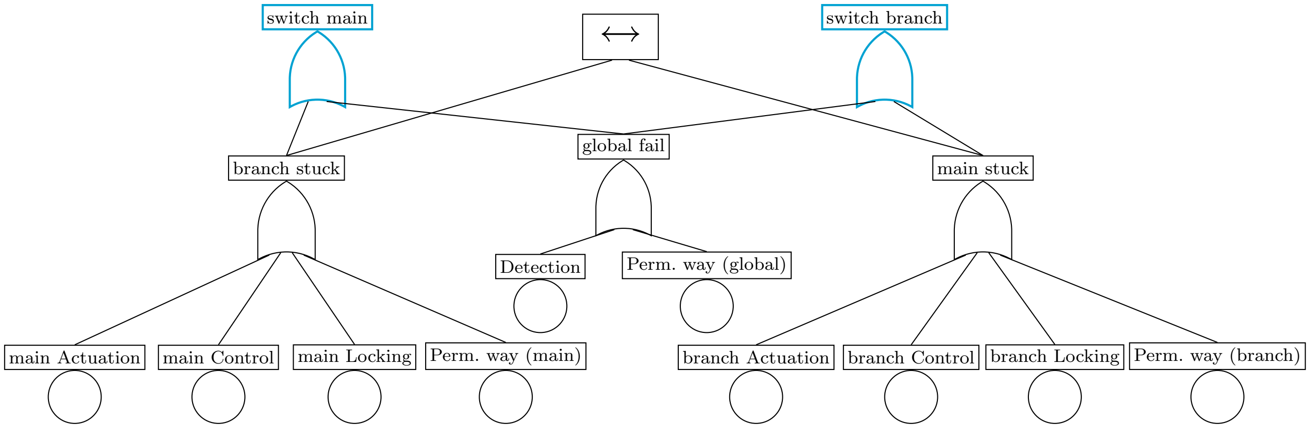


Fig. 7 Switch fault tree

switch can only be stuck in one position and not in both positions at once.

For the underlying failure modes, we use the categorization of technical switch failures presented in [6] for the UK rail network. Five different failure categories were identified:

- *Actuation* (A): failures in the track switching process, e.g., blade movement, lock actuation,
- *Control/Power* (C): failures in control or power supply of switch subsystems,
- *Detection* (D): failure to detect/transmit the position of switch rails/locks,
- *Locking* (L): failure to lock the switch blades, and
- *Permanent Way* (P): mechanical failures of rails, stretcher bars, slide chairs, etc.

Failures in the detection (D) or transmission of the current switch position or locks render the complete switch unusable and belong to global fail. Failures in the permanent way (P) can render only one of the tracks or both tracks unusable. Global failures (Perm. way (global)) comprise, for instance, failures in the crossing or the ballast. Position-specific failures (Perm. way (main/branch)) encompass failures of the blade rail or the guiding rail.

The remaining three categories Locking (L), Control (C) and Actuation (A) are position-specific failures as they originate in the context of blade movement. For example, let the switch be in the branch position. A failure of the actuation (main Actuation means that the switch blades cannot be moved to the main position anymore. As a result, the switch is "stuck" in the branch track (branch stuck) and the main track of the switch becomes unavailable, i.e., switch main fails.

4.3.2 Slip switches

While switches offer two routing options, slip switches have two ingoing and two outgoing tracks and offer up to four different routing options. Intuitively, they can be thought of as combining two switches: one for the ingoing tracks and one for the outgoing track. For the ingoing tracks, we distinguish between the *right track* and the *left track*. Depending on the joint position of the switch blades, a train arriving on an ingoing track is either routed going straight through on the *main track* or uses the diverging *branch track*. Note that it may be the case that not all four routing options are realized in practice.

The DFT model for a slip switch is depicted in Fig. 8. It consists of four top events (right/left × main/branch) corresponding to the four different routing options. All four events might be connected to multiple train paths. Failure of the complete slip switch could be modeled by adding an OR-gate with the four directions as children.

We model the slip switch as consisting of two switches (switch 1 and switch 2). If any of the two switches encounters a complete failure (switch 1/2 global fail), all four directions are rendered unusable. Both elements for the global failure are therefore connected to all four elements corresponding to the directions.

A partial failure occurs if one of the two switches is stuck in one direction. In this case, only two tracks become unavailable. This is modeled by the four elements switch 1/2 right/left stuck which each render two directions unusable. As before, we are using two MUTEXes to ensure that the blades are only stuck in position.

For presentation purposes, we do not depict the BEs in the slip-switch DFT. However, they are similar to the ones in the single switch DFT. For example, each of the elements for stuck position (switch 1/2 right/left stuck) has four BEs as children for Actuation, Control, Locking and Perm. way.

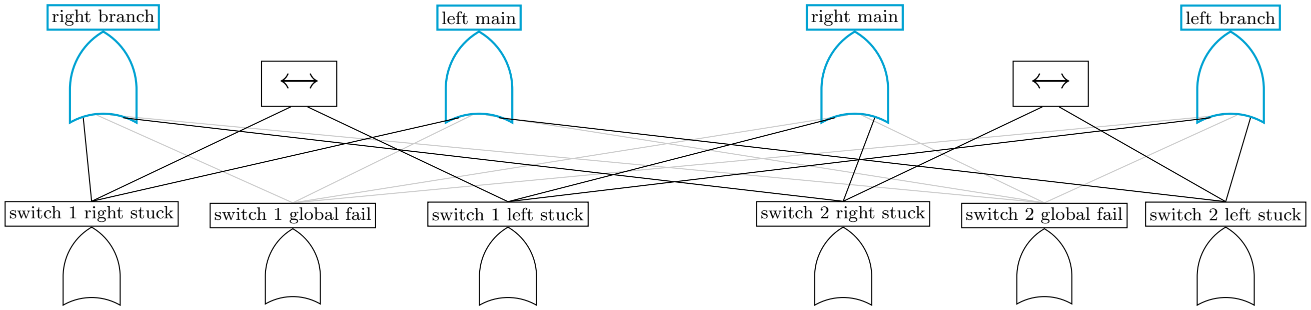


Fig. 8 Slip switch fault tree (without BEs)

4.3.3 Crossings

Crossings allow the overlapping of two tracks. However, in contrast to slip switches, no switching of the tracks is possible. As crossings have no electromechanical components such as a motor or switch blades, only failures of the permanent way are possible. The corresponding DFT is depicted in Fig. 9a and consists of an OR-gate with a single BE.

4.3.4 Further components

We focus on train routability in our DFT models and therefore model switches and slip switches with greatest detail. Further components such as signals or train detection are modeled as atomic components with less detail. If desired, the modular approach allows to refine the corresponding fault trees and model the failure behavior in greater detail. Using DFTs then also allows greater modeling flexibility by using the complete range of dynamic gates [25]. The fault trees for further components are depicted in Fig 9.

Track clearance detection. Track clearance detection checks whether the current track segment is occupied by a train. In Germany, axle counters are predominantly used for this purpose. The corresponding DFT is given in Fig. 9b. Failure for axle counters are subdivided into permanent failures of the component, e.g., due to power loss, and transient failures where a train axle was not correctly detected. In case of transient failures functionality is quickly restored by a reset [37].

Signals. The DFT for signal failures is given in Fig. 9c and considers intrinsic failures of the signal. Other system malfunctions connected with the interlocking system can also prevent the signal from being switched to green. However, these are not intrinsic failures of the signal and not considered here. The most common failures for signals are failures of the wayside electronics [37].

Track segments. Track failures are mainly caused by wear from train operations or insufficient support by the track foundations. The corresponding DFT is depicted in Fig. 9d. As the track segments in stations are small and usually traveled at low speed, failure rates for the track segments are typically short, cf. Sect. 4.4.

4.4 Failure rates

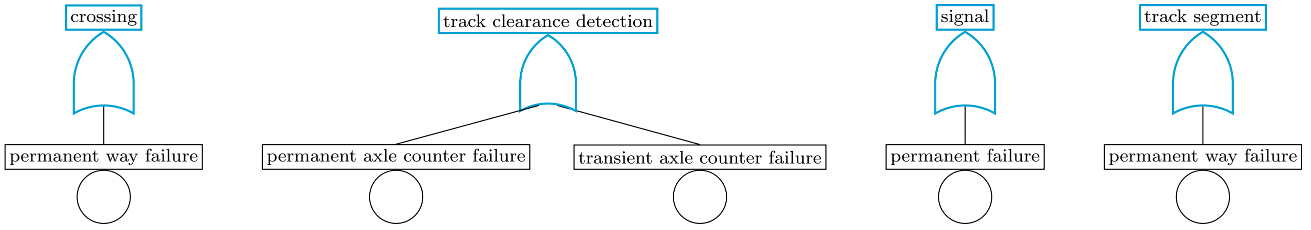
For the failure rates of BE elements, generic data derived from observed real-world failure statistics described in the literature is used. We summarize the failure rates for all infrastructure components in Table 1. Data sources and assumptions are described in the following.

4.4.1 Switches and crossings

The failure rates for the switch, slip switch and crossing are based on data from the UK Railway Network discussed and provided in [6]. We assume electromechanical actuation systems (Type HW/W63) for switches as they are the most widespread design in both the UK and German railway network. We took the failure rates from the MTTFRI data (mean-time-to-failure-requiring-intervention) for the different failure causes in [6, Table 4].

The three failure types Locking (L), Control (C) and Actuation (A) only occur during blade movements. Hassankiadeh [29] showed that almost 80% of the switch failure causes are due to blade obstruction, by, e.g., snow, ice, ballast, or insufficient lubrication of the slide chairs. These types of failures are more likely to occur when moving the blade to the position used less frequently. Thus, it would make sense to incorporate *load dependent* failure rates which take into account the number of trains traveling over both directions. However, no consistent information on the effects of load on the failure rates could be found in the literature. We therefore assume a uniform distribution over both branches. Nevertheless, load dependency could be incorporated in future work, for example based on degradation modeling of subsystems as in [49].

The five failure rates λ_A, λ_C, λ_D, λ_L, λ_P correspond to the previously presented failure causes Actuation (A), Control (C), Detection (D), Locking (L) and Permanent Way (P), respectively. As seen before, Permanent Way (P) failures can either affect the complete switch or only one of the branches.



(a) Crossing **(b)** Track clearance **(c)** Signal **(d)** Track segment

Fig. 9 Fault trees for further components

Table 1 Failure rates (in failures per day) for infrastructure components

Switches						Track segments	Signals	Axle counters	
λ_P	λ_A	λ_C	λ_D	λ_L	$\eta_{P,G}$	Failure (per km)	Failure	Reset request	Failure
1.46E–4	4.98E–4	2.26E–4	2.32E–4	1.28E–4	0.11	4.4E–4	2.9E–4	2.8E–4	1.1E–4

We use $\eta_{P,G}$ to denote the share of permanent way failures which render the complete switch failed. This factor is estimated based on the share of ballast, crossing, fishplate, and sleeper failures—leading to complete failure—in the UK failure cause data for switches provided in [29, Table 1].

The failure rates for slip switches and crossing use the same parameters. For slip switches, failures of type (A), (C), (D) and (L) apply to both switching motors, independently. For crossings, only failures of type (P) are considered.

4.4.2 Further components

For track segments and signals, failures rates are estimated based on the number of reported failures and the approximate number of elements in the UK network as given in [45]. The obtained failure rates are therefore consistent with the switch failure rates. For track clearance detection, no data could be found for the UK network–here, [37] can be used as an indication.

5 Quality metrics

We introduce the quality metrics used for the analysis in the following. The metrics are specified as properties in *continuous stochastic logic (CSL)* with reward extensions [3]. The properties use the labels of the CTMC to identify states in the CTMC. The atomic labels Failed(v) indicate states where the DFT node v has failed. For example, the label Failed(station) is added to each CTMC state where the complete railway station is considered failed. More complex properties can be built by Boolean combination of atomic labels.

In our setting, most metrics can be reduced to the *reachability probability* $\mathsf{P}^s(\Diamond^{\leq t}\ label)$ of reaching a state satisfying labeling $label$ from state s within time bound t.

5.1 Metrics for railway reliability modeling

We introduce the relevant metrics in the context of railway reliability modeling. The corresponding model-checking queries are formalized in Table 2.

5.1.1 General metrics

The most common metrics for fault trees are the *unreliability* and *mean-time-to-failure (MTTF)* of the system. The unreliability is formalized as the reachability probability of the top-level event station from initial state s_0 within time t. The MTTF is computed as the expected time ET of the failure of station.

Instead of the top-level event station, it is also possible to use any other DFT element instead. For example, the probability that a specific train type cannot be routed anymore at time t can be computed by the unreliability of the corresponding route set i. Similarly, the unreliability (or MTTF) of a train route route i or train path tp i can be computed.

5.1.2 Re-routing probability

The probability that at least one train must be re-routed can be computed by slightly adapting the DFT. Instead of having multiple routes per route set, only the scheduled route is considered. As a result, station fails as soon as a scheduled route becomes unavailable. Computing the unreliability of the overall system then yields the re-routing probability.

5.1.3 Criticality of infrastructure elements

An important metric in the DFT model for the station area is the *criticality* of infrastructure elements. The criticality denotes the influence of failures of a specific field element on the overall unreliability of the station area. Failure of a

Table 2 Model-checking queries

Measure	Model-checking query
Unreliability	$\mathsf{P}^{s_0}\left(\lozenge^{\leq t}\,\mathsf{Failed(station)}\right)$
MTTF	$\mathsf{ET}^{s_0}\left(\lozenge\,\mathsf{Failed(station)}\right)$
Unreliability for route i	$\mathsf{P}^{s_0}\left(\lozenge^{\leq t}\,\mathsf{Failed(route\ i)}\right)$
Unreliability for train path i	$\mathsf{P}^{s_0}\left(\lozenge^{\leq t}\,\mathsf{Failed(tp\ i)}\right)$
Criticality of component v	$\widetilde{I}_v(t)$
Unreliability after component v failed	$\sum_{s\in S,\mathrm{Failed}(v)\in L(s)} \mathsf{P}^{s_0}\left(\neg\mathrm{Failed}(v)\ \mathsf{U}\ s\right)\cdot \mathsf{P}^{s}\left(\lozenge^{\leq t}\,\mathsf{Failed(station)}\right)$
MTTF after component v failed	$\sum_{s\in S,\mathrm{Failed}(v)\in L(s)} \mathsf{P}^{s_0}\left(\neg\mathrm{Failed}(v)\ \mathsf{U}\ s\right)\cdot \mathsf{ET}^{s}\left(\lozenge\,\mathsf{Failed(station)}\right)$
Risk Achievement Worth for component v	$\mathsf{Unr}^t_{F[v\text{ is always failed}]}(\mathsf{Failed(station)})\ /\ \mathsf{Unr}^t_F(\mathsf{Failed(station)})$

highly critical element will in most cases lead to a failure of the complete station area, whereas uncritical elements have negligible influence on the overall unreliability. One common approach for criticality assessment is the computation of the *Birnbaum importance index* [9]. Let the *unreliability* of a set of states e from initial state s_0 at time t be denoted by the transient probability $\mathsf{Unr}^t(e) = \mathsf{P}^{s_0}(\lozenge^{=t}\, e)$. The Birnbaum index for component v at time t is defined as:

$$I_v(t) = \frac{\partial \mathsf{Unr}^t(\mathsf{TLE})}{\partial \mathsf{Unr}^t(v)}$$

and measures the influence of changing the component unreliability on the overall system unreliability. A Birnbaum index close to 1 indicates a high criticality of the component, whereas a value close to 0 indicates that a failure of the component has no influence on the system. A negative value indicates that a component failure decreases the overall system unreliability.

Obtaining the Birnbaum index $I_v(t)$ is computationally expensive, and we use the approximation from [46]:

$$\widetilde{I}_v(t) = x \cdot \left(\frac{\mathsf{Unr}^t(\mathrm{Failed}(\mathsf{TLE}) \wedge \mathrm{Failed}(v)))}{\mathsf{Unr}^t(\mathrm{Failed}(v)))} - \frac{\mathsf{Unr}^t(\mathrm{Failed}(\mathsf{TLE}) \wedge \neg\mathrm{Failed}(v)))}{\mathsf{Unr}^t(\neg\mathrm{Failed}(v))}\right)$$

$$\text{with } x = \frac{\mathsf{Unr}^t_F(\mathrm{Failed}(v))}{\mathsf{Unr}^t_{F_{iso}}(\mathrm{Failed}(v))}.$$

The approximation intuitively computes the difference between the system failure while the component is failed and the system failure while the component is still available. The factor x computes the fraction of the unreliability of the component *in the system* F and *in isolation* F_{iso}. The DFT F_{iso} for component v is obtained from F by setting $\mathsf{TLE}(F_{iso}) = v$ and removing all restrictions, e.g., removing all MUTEX.

5.1.4 Further metrics

Probabilistic model checking allows to compute further—more complex—metrics as well. For instance, it is important to estimate the time frame in which a failed element should be repaired or replaced. To this end, we compute the *unreliability after component failure*. For component v we consider each state s in the CTMC, where v is newly failed, i.e., v is operational in all predecessors of s. Using state s as initial state, we compute the unreliability of the overall system and use a typical maintenance interval as time bound. The result is scaled with the probability to reach state s in the first place. Similarly, the *MTTF after component failure* can be calculated. For both computations, we use the improved algorithm from [25], which only requires two model checking queries: (1) the probability to reach each state s, and (2) the unreliability/MTTF starting in s.

Apart from the Birnbaum index, other criticality metrics can be analyzed as well. One example is the *risk achievement worth (RAW)* [58] which gives the impact of immediate failure of component v on the system unreliability. It is calculated by computing the unreliability of the system at time point t in both the original DFT and a modified DFT where component v is replaced by one which is always failed. The fraction of both results determines the RAW.

6 Evaluation

6.1 Input data

As input data, we rely on infrastructure and timetable data for the German railway network from 2014. The timetable data we use are mainly based on passenger traffic and some regular freight services. Short-notice freight services and shunting movements are not accounted for.

The data are read from exchange formats used by German infrastructure manager DB Netz AG. As access to control tables encoded in safety-critical interlocking is highly restricted, we employ an approach similar to the one discussed in [43] to generate the correspondence between train routes and the required state of field elements. The approach explores the infrastructure graph and the possible train paths between station boundaries and holding positions. Of course, this approach can only provide an approximate description of the safety logic projected in the interlocking, as, e.g., not all feasible routes need be projected in interlocking. Still, we deem it sufficient to analyze the infrastructure and usability of our method. In case real interlocking control tables are available, the method can easily be adapted.

6.2 Infrastructure considered in the analysis

We evaluate our approach on four different railway stations in the German state North Rhine-Westphalia: Aachen Hbf, Mönchengladbach Hbf, Wuppertal Hbf and Herzogenrath Bf. The former three stations are major central stations with multiple starting and ending train lines. Aachen Hbf consists of 9 tracks (7 platform tracks), Mönchengladbach Hbf has 10 tracks (9 platform tracks) and Wuppertal Hbf has 5 tracks (5 platform tracks). Herzogenrath Bf is a smaller medium size station with 4 platform tracks (3 in use) and a small freight yard.

In our analysis, we focus on switches and do not consider track segments, signals or axle counters. Including the other three elements is perfectly viable from a computational point of view and does not require any changes in the implementation. However, we concentrate on switches for four main reasons:

- Switches are the most interesting component from a routability perspective. They experience various modes of degradation where specific directions are unusable—by the blades being “stuck” in one direction—while other routing options are still available.
- Switch failures have been shown to be one of the most important factors in delay build-up [6] and have been the continuous focus of research on design and asset monitoring improvements in reliability engineering.
- Compared to rails, switches are more complex and vulnerable and, hence, fail significantly more often than track segments. In addition, design specifications of railway line segments are fixed, such that reliability can mainly be improved by shortening inspection intervals.
- Failures of signals and axle counters tend to yield milder disruptions compared to switch failures.

6.3 Set-up

We use Storm[59] in version 1.6.3 as back-end for our DFT analysis. We run the evaluation on a Linux machine restricted to 32GB RAM and use a single Intel Xeon Platinum 8160 processor with 2.1GHz.

We employ the workflow presented in Fig. 1. From the given infrastructure data for the four railway stations, we automatically generate DFT models. We analyze the DFTs with Storm according to the metrics presented in Sect. 5. For presentation purposes, we focus on the general metrics (unreliability and MTTF) and the Birnbaum importance index. For each railway station, we consider two different route sets:

- *sched*: each route set only contains the scheduled route. The unreliability then corresponds to the re-routing probability.
- *alt 5*: each route set contains the five most feasible routes according to the priorities in the input data.

We also consider two different levels of detail in which we model the infrastructure components:

- *single*: each component is modeled by a single basic event. The resulting model is a *static* fault tree, because no MUTEX is present.
- *refined*: detailed DFTs are used for the components, in particular for switches (cf. Fig. 7) and slip switches (cf. Fig. 8).

The generated DFT models are publicly available (in anonymized form) on our website[1].

7 Results

7.1 Model characteristics

We give the characteristics of the considered scenarios in Table 3. Each scenario is specified by the railway station, the considered routing (scheduled route or 5 alternative routes) and the level of detail for components (single BE or refined DFT). We identify each scenario with a unique id. Columns five to eight provide the number of route sets, the number of routes, the number of train paths and the number of physical components (switches, slip switches and crossings) in the station area. The size of the resulting DFT is provided in the last three columns. We give the number of basic events and the number of static and dynamic gates.

Model sizes. The models of the railway stations of Aachen and Mönchengladbach are the largest as they contain the largest number of infrastructure elements. Wuppertal is considerably smaller—mainly because it only has 5 platform tracks and is a through station on a line. Herzogenrath is the

[1] http://www.stormchecker.org/publications/dfts-for-railway-stations.

Table 3 Model characteristics

Scenario				Railway				DFT nodes		
Id	Station	Routing	Detail	Route sets	Routes	Train paths	Comp.	BE	Static	Dynamic
1	Aachen	Sched	Single	59	59	44	54	54	325	0
2			Refined					545	438	54
3		Alt 5	Single	14	66	29	46	46	248	0
4			Refined					464	344	46
5	Herzogenrath	Sched	Single	11	11	13	22	22	96	0
6			Refined					194	135	19
7		Alt 5	Single	10	36	25	25	25	141	0
8			Refined					224	186	22
9	Mönchengladbach	Sched	Single	30	30	31	41	41	229	0
10			Refined					481	333	48
11		Alt 5	Single	11	55	41	47	47	259	0
12			Refined					523	371	52
13	Wuppertal	Sched	Single	26	26	23	27	27	163	0
14			Refined					300	226	30
15		alt 5	Single	14	49	28	27	27	179	0
16			Refined					300	242	30

smallest model with 3 platform tracks. The resulting DFTs contain up to 500 BEs and a thousand elements in total, placing them among the largest DFTs in the literature [51]. One characteristic of the resulting DFTs is that they are highly coupled as switches can be linked to several train paths. This makes the analysis computationally harder as no subtrees can be analyzed individually.

7.2 Results for station failure

We start by analyzing the failure of the overall station. A station is considered failed if at least one type of train cannot be routed anymore. We provide the corresponding analysis results in Table 4.

We refer to a scenario by its id. The starred ids refer to a scenario where we restrict our analysis to a maximal number of 4 consecutive failures. In practice, more than a few consecutive failures do not occur, as repairs or replacements will take place before. It is therefore reasonable to restrict our analysis to 4 consecutive failures. We use this restriction to mitigate the state-space explosion problem for scenarios 4 and 12 which otherwise result in a memory out (MO).

For each considered DFT, we give the size of the resulting CTMC in terms of states and transitions (columns 2 and 3) as well as the time (in seconds) required to build the CTMC (column 4). We analyze the model with respect to two measures (cf. Sect. 5): the unreliability within 90 days—which is a typical maintenance interval—and the MTTF (in days). The last column indicates the total time in which both model checking queries were computed.

CTMC sizes. The size of the generated CTMCs ranges from 2 to nearly 6 million states. When only considering the scheduled route and single BEs, the CTMC only contains two states—initial and failed—as each failure directly renders one route unavailable. In case of refined subtrees, the CTMCs are still quite small as most failures lead to a complete failure as well. In contrast, when considering multiple alternative routes, the failure of one route can still be mitigated by using an alternative route. When considering single BEs, the resulting CTMCs are still relatively small, because the corresponding DFTs contain only up to 50 BEs and—more importantly—no dynamic gates (cf. Table 3). For the refined DFTs, however, the resulting CTMCs become intractable, yielding a memory out for Aachen and Mönchengladbach. In these cases, we restricted our state space generation to only explore up to 4 consecutive failures. This reduces the state-space size considerably and allows to analyze these stations as well.

Analysis results. Table 4 indicates that the unreliability of a station area after 90 days is nearly 1 when only considering the scheduled route—except for Herzogenrath where the unreliability is slightly better. This means that the system will almost certainly experience at least one service-affecting failure on the station level within 90 days. The corresponding MTTF lies between 15 and 52 days. Given the fact that Aachen, for instance, features 54 switches and crossings, each one having an MTTF of about 2 years, the results seem plausible. The high unreliability does not necessarly reflect the observed effects in practice, as (1) failed field elements

Table 4 Analysis results

		CTMC construction			Model checking queries		
	Id	States	Transitions	Time (s)	Unreliability	MTTF (d)	Time (s)
Aachen	1	2	2	0.00	0.997	15.04	0.00
	2	2049	13,313	0.43	0.996	16.38	0.01
	3	769	5329	0.15	0.913	36.94	0.15
	4	–	–	MO	–	–	–
	4*	1,174,596	5,891,462	103.61	0.784	57.09	2.22
Herzogenr.	5	2	2	0.00	0.879	42.69	0.00
	6	257	1281	0.01	0.826	51.54	0.00
	7	232	1489	0.02	0.704	73.86	0.00
	8	13,801	153,049	2.75	0.495	127.70	0.12
	8*	8636	61,743	1.06	0.496	124.62	0.04
M'gladbach	9	2	2	0.00	0.995	16.94	0.00
	10	8193	61,441	1.25	0.991	19.01	0.05
	11	22,658	228,251	3.45	0.867	47.81	0.22
	12	–	–	MO	–	–	–
	12*	5,912,302	32,950,979	480.08	0.692	72.37	15.07
Wuppertal	13	2	2	0.00	0.964	27.11	0.00
	14	65	257	0.01	0.953	29.50	0.00
	15	312	1637	0.03	0.855	47.04	0.00
	16	145,925	1,631,261	36.28	0.612	89.64	1.55
	16*	44,219	273,656	5.11	0.617	86.22	0.15

will be repaired or replaced, and (2) re-routing of trains is possible.

The positive effects of redundancy can be seen from the scenarios where re-routing to alternative routes is considered. Here, the unreliability decreases significantly. When considering refined DFTs, the MTTF triples for all stations except Herzogenrath. The exception can possibly be explained by the fact that Herzogenrath, unlike the other stations, features two single-track railway lines with scheduled operations. As a result, no re-routing is possible for trains entering from or exiting to those lines in case the outermost switch fails.

When comparing the results of a restricted exploration (starred ids) to an unrestricted one, we see that the values for the unreliability and MTTF differ by roughly 2% for Herzogenrath and up to 4% for Wuppertal. Thus, we still obtain insightful results even for restricted exploration. It is also important to note that the restricted exploration provides worst-case results and the exact result will always be better.

7.3 Criticality analysis

Results for the criticality analysis of switches, slip switches and crossings are provided in Table 5.

We compute the criticality for the complete switch/slip switch when using single BEs for components. In case of the refined DFTs for components, we compute the criticality for each of the two (four) branches of the switch (slip switch). For crossings, we always compute the criticality for the complete component.

The first column references the id of the considered scenario. The second column gives the total number of elements for which criticality was computed. In the variant with single BEs, this number coincides with the number of components in the railway station.

We evaluate two different approaches for the criticality analysis: BDD-based and via model checking. For the scenarios with single BEs, the resulting model is a static fault tree (SFT) and we can use an approach based on BDDs to compute the Birnbaum importance index [19]. The third column indicates the number of nodes in the corresponding BDD, and the fourth column gives the total time required to analyze all components with the BDD approach. Note that for DFTs, the BDD-based approach is not applicable (n.a.). The model checking approach is applicable to all scenarios, and columns five to seven present the corresponding results. Column five indicates the number of states in the resulting CTMC, and column six gives the time (in seconds) it took to analyze an element. Note that we calculate the criticality for all elements (switches, slip switches, crossings) independently and therefore provide intervals of minimal and maximal values (over all elements) for both columns. The seventh column indicates the total time required to compute the criticality for all elements.

Table 5 Results for criticality analysis of switches

			BDD-based		Model checking-based			Criticality
	Id	Elem.	Nodes	Tot. time [s]	States	Time [s]	Tot. time [s]	Results
Aachen	1	54	55	0.98	[2, 3]	[0.03, 0.03]	1.49	[0.0025, 0.0031]
	2	113	–	n.a.	[2048, 6144]	[2.73, 5.07]	551.57	[−0.0021, 0.0047]
	3	46	22	1.00	[769, 1282]	[1.14, 1.81]	63.26	[0.0000, 0.1092]
	4	96	–	n.a.	–	–	MO	–
	4*	96	–	n.a.	[1,109,775, 1,432,106]	[643.96, 789.21]	68,203.81	[0.0535, 0.2512]
Herzogenr.	5	22	23	0.66	[2, 3]	[0.02, 0.02]	0.37	[0.1216, 0.1515]
	6	42	–	n.a.	[257, 768]	[0.24, 0.28]	10.52	[−0.0079, 0.1903]
	7	25	25	0.84	[232, 365]	[0.23, 0.31]	6.50	[0.0000, 0.3690]
	8	48	–	n.a.	[13,801, 22,880]	[18.17, 30.45]	1098.22	[−0.1185, 0.5631]
	8*	48	–	n.a.	[8636, 12,184]	[6.30, 12,184]	351.56	[−0.1070, 0.5628]
M'gladbach	9	41	42	0.91	[2, 3]	[0.02, 0.04]	0.96	[0.0049, 0.0061]
	10	97	–	n.a.	[8192, 24,576]	[6.84, 13.63]	1260.52	[−0.0039, 0.0101]
	11	47	1,033	1.09	[22,658, 34,970]	[29.05, 46.82]	1645.05	[0.0000, 0.1665]
	12	107	–	n.a.	–	–	MO	–
	12*	107	–	n.a.	[5,486,213, 6,956,578]	[2523.45, 2991.12]	297,626.65	[0.2236, 0.3926]
Wuppertal	13	27	28	0.76	[2, 3]	[0.02, 0.03]	0.52	[0.0404, 0.0451]
	14	60	–	n.a.	[65, 192]	[0.33, 0.37]	20.52	[−0.0120, 0.0525]
	15	27	52	1.32	[302, 408]	[0.37, 0.46]	10.36	[0.0000, 0.1815]
	16	60	–	n.a.	[145,925, 259,200]	[312.31, 614.00]	25,762.23	[−0.1021, 0.4307]
	16*	60	–	n.a.	[44,219, 62,996]	[40.48, 53.39]	2764.00	[−0.0661, 0.4283]

The criticality results as given by the Birnbaum importance index are provided in the last column. We again provide the interval of minimal and maximal values over all considered elements. Naturally, the obtained criticality result for each element is the same for both the model checking-based and the BDD-based approach.

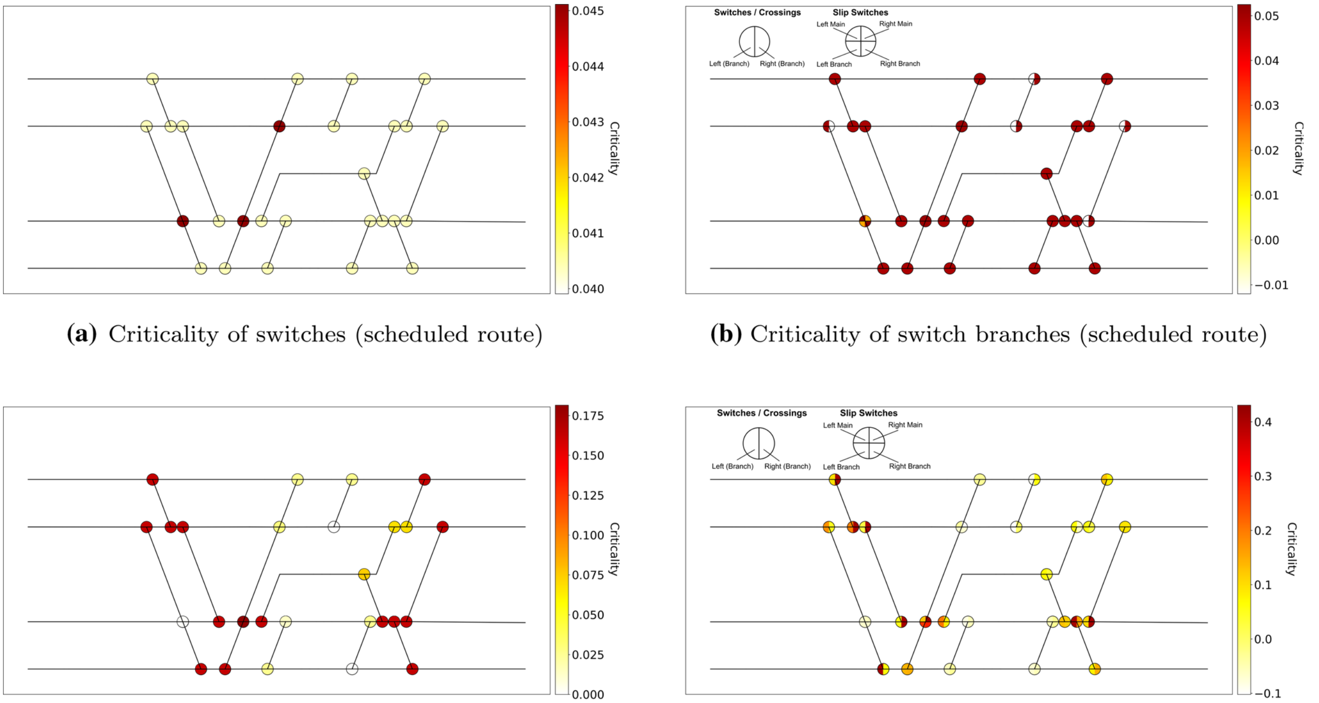
Negative criticality values. We found that, interestingly, some switch configurations in the refined model exhibit slightly negative criticality values. This indicates situations where failures of specific switch branches *improve* the overall system reliability (cf. Sect. 5.1.3). Verifying with train routing options, we found that the corresponding branches are not used by any train. As a result, this type of branch failure does not have any negative effects on train operations. On the contrary, the MUTEX in the DFT model ensures that a failure of such an "irrelevant" switch configuration—which turns the switch "stuck" in the other configuration—reduces the probability of failures of the other configuration as no blade movements are performed any more. Thus, a failure of one switch configuration can improve the reliability of the other switch configuration—which is indicated by the negative criticality values.

Visualization of criticalities. We also visualize the Birnbaum importance index for all elements at time point 90 days in Fig. 10.

We focus as an example on Wuppertal Hbf for a detailed presentation, but comparable plots are possible for the other stations as well. We provide the criticality results for both route sets with only the scheduled route (upper row) and route sets with 5 alternative routes (lower row). We again consider switches either as a single entity (left column) or distinguish between the different branches (right column). Each colored dot represents a switch, slip switch or crossing where red indicates a higher and yellow a lower relative criticality. We use 2 (4) segments inside each colored dot to distinguish between the different branches for each switch (slip switch), where left/right refers to the directionality seen from the element's narrow end.

In addition, we visualize the criticality results for Mönchengladbach Hbf in Fig. 11. We consider the scenario with 5 alternative routes and the refined switches. Grey dots indicate switches which are not amongst the 5 train routes with highest priority for any train type.

Discussion. Comparing the criticality results in Fig. 10, we see that for the scheduled route with single BEs (Fig. 10a), all switches have the same criticality as each failure directly leads to a complete failure. Slip switches, due to their more complex design, are more failure-prone and consequently more critical. For the refined DFTs (Fig. 10b), a similar picture holds true. However, here it can be seen that some switch branches are non-critical. This can be explained by the fact



(a) Criticality of switches (scheduled route)

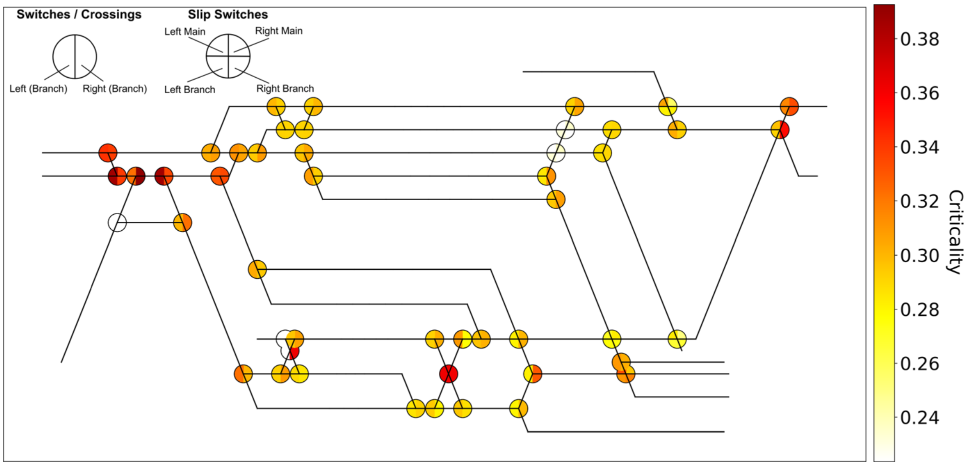
(b) Criticality of switch branches (scheduled route)

(c) Criticality of switches (alternative routes)

(d) Criticality of switch branches (alternative routes)

Fig. 10 Criticality of switches in Wuppertal Hbf

Fig. 11 Criticality of switch branches (alternative routes) in Mönchengladbach Hbf

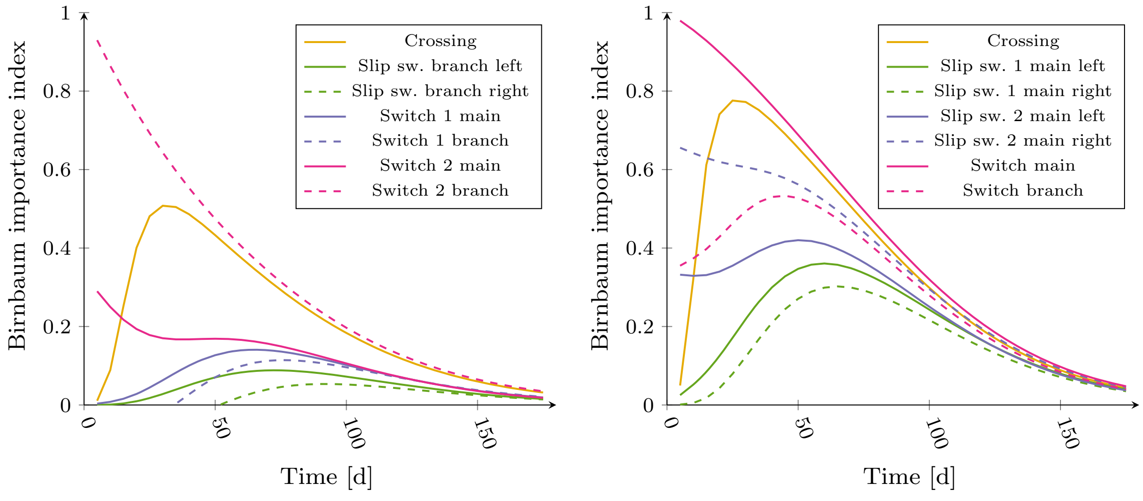


that these branches are not used by any train path in scheduled operations.

A more nuanced picture unfolds when taking alternative routes into account. The criticality results in Fig. 10c, d as well as in Fig. 11 show that switches in more extremal positions on ingoing and outgoing tracks tend to be more critical than switches centrally located in the station. Please note that we do consider and allow for re-routing to both tracks on adjacent lines, in case corresponding routes are projected in the infrastructure data. This mitigates situations where a single switch connecting to the adjacent line can have a totally blocking effect. Still, the redundancy in terms of the number of parallel routing options increases in the central (platform) area of the station and leaves extremal switches more critical.

Criticality over time. The criticality values so far have all been computed for the same time point of 90 days. In Fig. 12, we depict the criticality values for different time points from 5 days up to 6 months. We use the refined DFTs and the alternative routing options to present the criticality for selected components in Aachen Hbf (Fig. 12a) and Mönchengladbach Hbf (Fig. 12b).

We see that the criticality values are heavily influenced by the considered time point. As time progresses, different components are among the most critical. This is mainly due



(a) Aachen Hbf **(b)** Mönchengladbach Hbf

Fig. 12 Criticality of components for different time points

to the different failure rates of the considered components and also the overall degradation of the system. The crossing for example starts with a very low criticality, reaches its most critical value around 30 days and afterwards becomes less critical again. A possible reason for this finding is that crossings, having no movable items, exhibit the lowest failure rates. For very short intervals, they are unlikely to fail and, hence, less critical than other elements. For longer time horizons, they start to contribute, and given other elements might have already failed, become a significant factor for the remaining system performance featuring a reduced set of available routes. Similar, yet less drastic behavior can be seen for slip switch 1 in Fig. 12b as well, possibly indicating that it becomes relevant in ensuring residual system performance. In general, the criticality of all components converges to zero, as further failures have no impact any more when the routing has failed already.

8 Discussion

We discuss the practical implications obtained from the results and also discuss our approach in general.

8.1 Practical implications

The reliability analysis performed in this paper shows a very high probability of encountering at least one service-affecting failure in a period of 90 days. From this finding—together with a MTTF in the range of 15–50 days for the fault tree only considering scheduled routes—it becomes clear that scheduled train routes need to be re-arranged on a regular basis to account for disruptions. As a result, it can be stated that considering and accounting for line planning alternatives in infrastructure and timetable planning is a *must* to ensure high quality train operations.

The *alt 5* scenarios, where the 5 most suitable routes for each train are considered, yield insight into the flexibility and resilience of the infrastructure. The quality of this heuristic cut-off—motivated by model performance aspects—depends on the size of the station, the number of routing options given in the input data, and the prioritization of the different train paths. For large stations, the alt5 sets are a subset of all topologically feasible train routing options. However, 3–5 routing alternatives should, in general, be sufficient to grasp the performance of the system and the criticality of the assets.

The criticality analysis performed in this paper provides assistance in determining which switches are most critical for train service operations. The switches which were found to be most critical could, for example, receive special focus in maintenance and investment planning to be replaced by more reliable components. Another option would be to closely monitor these switches such that degradations can be detected and preventive maintenance can be performed. In the analysis, switches at station entry/exit were typically found to be more critical than switches more centrally located in the station area. This would indicate that these switches should be particularly considered. On the other hand, switches leading to depot areas are found to be uncritical, which shows a limitation of our criticality analysis as shunting movements are not covered in the timetable data. Still, in order to obtain a complete picture, corresponding train movements should be addressed, as well.

8.2 Model comparison

Computation times. Most of the DFT analysis can be performed within seconds (cf. Table 4). The only exceptions are the refined DFTs with alternative routes for stations Aachen and Mönchengladbach. The corresponding CTMCs consist of roughly 1.2 million and 6 million states, respectively. Constructing the 6 million states can be done within 8 minutes, which is still a reasonable time. After constructing the state space, the actual model checking can be performed within seconds. Even the CTMC with 6 million states can be analyzed within 15 seconds. Our approach constructs the CTMC only once and allows to efficiently compute a variety of measures without the need to rebuild the CTMC.

For the criticality analysis, the size of the CTMC is again a major factor for the computation time. Additionally, we compute the criticality for each component and therefore need to construct a dedicated CTMC for each component. As a result, computing all criticalities takes a long time (cf. Table 5). For instance, the analysis for Mönchengladbach took over 3 days. On the plus side, each component can be analyzed individually and the complete analysis can be parallelized very easily.

Using BDDs for the criticality scales significantly better than using model checking (cf. Table 5). While for example model checking required over 27 min for scenario 11, the BDD-based approach could compute the criticality results in one second. However, analysis via BDDs is limited to static fault trees which are not as expressive and therefore not as detailed as DFTs.

The criticality of components can significantly change over time (cf. Fig. 12). Some components are critical in the beginning of the operation while other components become more important during operation. It is therefore crucial to consider the evolution of the criticality values over time to make fully informed decisions.

Level of modeling detail. While the models with single BEs can be analyzed very quickly, the computed results are not as meaningful as for the refined DFTs. Both the unreliability and the MTTF are more pessimistic for the single BEs than for the refined one. The created SFTs are conservative compared to the DFTs, as the MUTEX in the DFT restricts the possible failures. In fact, we obtain better reliability results without modifying the infrastructure but just through more detailed modeling. Refining existing models might therefore be a good choice to obtain a more accurate view on the current infrastructure before planning any new investments.

For the criticality, the values computed on the DFT have greater variation than the ones for the SFT. This difference between a detailed model and a rough model is best visualized in Fig. 10. When considering the alternative routes (bottom row), the highly critical components are located at the ingoing and outgoing tracks. For the single BEs (Fig. 10c), all these components have nearly the same criticality. While this still holds true when considering the refined model (Fig. 10d), more nuanced insights are possible. For instance, it is clearly visible that each of the switches has a branch that is more critical than the other one. This branch is used more often for train operations and therefore more critical. Similar conclusions can be made for the criticality of switches in Mönchengladbach (Fig. 11).

9 Conclusion and outlook

In this paper, we presented a DFT model to analyze railway infrastructure reliability. By focusing on train routability and train paths projected in the signaling system, a service-centered view is adopted, which allows to assess infrastructure failures w.r.t. their implications on train operations. The fault trees obtained in the analysis of major stations contain up to 6 million states and are amongst the largest described in the literature.

By allowing to pinpoint critical infrastructure components, the approach is suited to provide assistance in asset management to improve the effectiveness and efficiency of infrastructure investments, maintenance and monitoring systems. In its present form, our modeling approach's main area of use is thought to be comparative assessment of wayside infrastructure elements to better understand their role and influence on operations and to adjust investments accordingly. The focus has been on scheduled routes for passenger traffic, such that the results could be used to review train routes in timetable planning. In future, especially when reviewing dismantling of switches to reduce failure frequencies, extensions also fully accounting for freight service and shunting movements are desirable to provide a more realistic picture on the use and necessity of switches.

Another line left for future research would be to further improve the representation of infrastructure component failures by introducing a more detailed representation of the effects of operations on failure processes on the BE-level. By incorporating the effects of traffic load, train masses, or the number of blade movements the predictive quality of the DFT model for failure processes and reliability assessment could be further strengthened. As a result, the application area of the methodology could be further extended beyond comparative analysis of infrastructure components and be used for requirement-based system design.

Acknowledgements The authors would like to thank Daniel Basgöze and Markus Miliats (both at RWTH Aachen University) for their contributions to the development of the DFT analysis in STORM.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The input data are not available due to confidentiality. The DFT models are publicly available at http://www.stormchecker.org/publications/dfts-for-railway-stations.

Code Availability The code can be made available upon request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Andrews, J., Prescott, D., Roziéres, F.D.: A stochastic model for railway track asset management. Reliab. Eng. Syst. Saf. **130**, 76–84 (2014)
2. Baier, C., de Alfaro, L., Forejt, V., Kwiatkowska, M.: Model checking probabilistic systems. In: Handbook of Model Checking, pp. 963–999. Springer (2018)
3. Baier, C., Hahn, E.M., Haverkort, B.R., Hermanns, H., Katoen, J.P.: Model checking for performability. Math. Struct. Comput. Sci. **23**(4), 751–795 (2013)
4. Basile, D., ter Beek, M., Fantechi, A., Gnesi, S., Mazzanti, F., Piattino, A., Trentini, D., Ferrari, A.: On the Industrial Uptake of Formal Methods in the Railway Domain, LNCS, vol. 11023, pp. 20–29. Springer (2018)
5. Basile, D., ter Beek, M.H., Ciancia, V.: Statistical model checking of a moving block railway signalling scenario with Uppaal SMC-experience and outlook. In: Proceedings of ISoLA, LNCS, vol. 11245, pp. 372–391. Springer (2018)
6. Bemment, S.D., Goodall, R.M., Dixon, R., Ward, C.P.: Improving the reliability and availability of railway track switching by analysing historical failure data and introducing functionally redundant subsystems. Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit **232**(5), 1407–1424 (2017)
7. Bešinović, N.: Resilience in railway transport systems: a literature review and research agenda. Transp. Rev. **40**(4), 457–478 (2020)
8. Biagi, M., Carnevali, L., Paolieri, M., Vicario, E.: Performability evaluation of the ERTMS/ETCS—level 3. Transp. Res. Part C **82**, 314–336 (2017)
9. Birnbaum, Z.: On the importance of different components in a multicomponent system. In: Multivariate Analysis-II, pp. 581–592 (1969)
10. Boudali, H., Crouzen, P., Stoelinga, M.: Dynamic fault tree analysis using input/output interactive Markov chains. In: Proceedings of DSN, pp. 708–717. IEEE (2007)
11. Budde, C.E., Ruijters, E., Stoelinga, M.: The dynamic fault tree rare event simulator. In: Proceedings of QEST, LNCS, vol. 12289, pp. 233–238. Springer (2020)
12. Butler, M.J., Körner, P., Krings, S., Lecomte, T., Leuschel, M., Mejia, L., Voisin, L.: The first twenty-five years of industrial use of the B-method. In: Proceedings of FMICS, LNCS, vol. 12327, pp. 189–209. Springer (2020)
13. CENELEC: EN 50128: Railway applications—Communication, signalling and processing systems–software for railway control and protection systems (2011)
14. CENELEC: EN 50129: Railway applications—communication, signalling and processing systems—safety related electronic systems for signalling (2017)
15. CENELEC: EN 50126-1/50126-2: Railway applications—the specification and demonstration of reliability, availability, maintainability and safety (RAMS) (2018)
16. Chen, S., Ho, T., Mao, B.: Reliability evaluations of railway power supplies by fault-tree analysis. IET Electr. Power Appl. **1**(2), 161–172 (2007)
17. Cimatti, A., Roveri, M., Tonetta, S.: Requirements validation for hybrid systems. In: Proceedings of CAV, LNCS, vol. 5643, pp. 188–203. Springer (2009)
18. Dugan, J.B., Bavuso, S.J., Boyd, M.A.: Fault trees and sequence dependencies. In: Proceedings of RAMS, pp. 286–293 (1990)
19. Dutuit, Y., Rauzy, A.: Efficient algorithms to assess component and gate importance in fault tree analysis. Reliab. Eng. Syst. Saf. **72**(2), 213–222 (2001)
20. Estevan, A.M.: Dependability and safety evaluation of railway signalling systems based on field data. Ph.D. thesis, Lulea University of Technology (2015)
21. Fantechi, A.: Twenty-five years of formal methods and railways: What next? In: SEFM, LNCS, vol. 8368, pp. 167–183. Springer (2013)
22. Ferrari, A., Magnani, G., Grasso, D., Fantechi, A.: Model checking interlocking control tables. In: FORMS/FORMAT 2010, pp. 107–115. Springer (2011)
23. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H., Fantechi, A.: Comparing formal tools for system design: a judgment study. In: ICSE '20, pp. 62–74. ACM (2020)
24. Fokkink, W., Hollingshead, P.: Verification of interlockings: from control tables to ladder logic diagrams. In: Proceedings of FMICS, vol. 98, pp. 171–185. CWI (1998)
25. Ghadhab, M., Junges, S., Katoen, J.P., Kuntz, M., Volk, M.: Safety analysis for vehicle guidance systems with dynamic fault trees. Reliab. Eng. Syst. Saf. **186**, 37–50 (2019)
26. Guck, D., Katoen, J.P., Stoelinga, M., Luiten, T., Romijn, J.: Smart railroad maintenance engineering with stochastic model checking. In: Proceedings of RAILWAYS. Civil-Comp Press (2014)
27. Gulati, R., Dugan, J.B.: A modular approach for analyzing static and dynamic fault trees. In: Proceedings of RAMS, pp. 57–63 (1997)
28. Hartonas-Garmhausen, V., Campos, S., Cimatti, A., Clarke, E., Giunchiglia, F.: Verification of a safety-critical railway interlocking system with real-time constraints. Sci. Comput. Program. **36**(1), 53–64 (2000)
29. Hassankiadeh, S.J.: Failure analysis of railway switches and crossings for the purpose of preventive maintenance. Master's thesis, KTH Stockholm (2011)
30. Henry, J.: Automatic fault tree construction for railway safety systems. Ph.D. thesis, Loughborough University (1996)
31. Hensel, C., Junges, S., Katoen, J.P., Quatmann, T., Volk, M.: The probabilistic model checker Storm. Int. J. Softw. Tools Technol. Transf. (2021)
32. Hermanns, H., Jansen, D.N., Usenko, Y.S.: From StoCharts to MoDeST: a comparative reliability analysis of train radio communications. In: WOSP, pp. 13–23. ACM (2005)

33. James, P., Roggenbach, M.: Automatically verifying railway interlockings using SAT-based model checking. Electr. Commun. EASST **35** (2011)
34. Junges, S., Guck, D., Katoen, J.P., Rensink, A., Stoelinga, M.: Fault trees on a diet: automated reduction by graph rewriting. Formal Asp. of Comput. pp. 1–53 (2017)
35. Junges, S., Guck, D., Katoen, J.P., Stoelinga, M.: Uncovering dynamic fault trees. In: Proceedings of DSN, pp. 299–310. IEEE (2016)
36. Junges, S., Katoen, J.P., Stoelinga, M., Volk, M.: One net fits all—a unifying semantics of dynamic fault trees using GSPNs. In: Proceedings of Petri Nets, LNCS, vol. 10877, pp. 272–293. Springer (2018)
37. Kalvakunta, R.G.: Reliability modelling of ERTMS/ETCS. Master's thesis, NTNU (2017)
38. Kanso, K., Moller, F., Setzer, A.: Automated verification of signalling principles in railway interlocking systems. Electronic Notes in Theoretical Computer Science **250**(2), 19–31 (2009). Proceedings of AVoCS
39. Kassa, E.: Analysis of failures within switches and crossings using failure modes and effects analysis methodology. In: Proceedings of Intelliswitch Symposium (2017)
40. Katoen, J.: The probabilistic model checking landscape. In: Proceedings of LICS, pp. 31–45. ACM (2016)
41. Katoen, J., Stoelinga, M.: Boosting fault tree analysis by formal methods. In: ModelEd, TestEd, TrustEd, *LNCS*, vol. 10500, pp. 368–389. Springer (2017)
42. Khaled, A.A., Jin, M., Clarke, D.B., Hoque, M.A.: Train design and routing optimization for evaluating criticality of freight railroad infrastructures. Transp. Res. Part B Methodol. **71**, 71–84 (2015)
43. Luteberget, B., Johansen, C.: Efficient verification of railway infrastructure designs against standard regulations. Formal Methods Syst. Des. **52**(1), 1–32 (2018)
44. Morant, A., Gustafson, A., Söderholm, P., Larsson-Kråik, P.O., Kumar, U.: Safety and availability evaluation of railway operation based on the state of signalling systems. Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit **231**(2), 226–238 (2017)
45. ORR-Office of Road and Rail: Online data portal, Rail infrastructure, assets and environmental. https://dataportal.orr.gov.uk/. Last accessed 01-05-2019 (2013)
46. Ou, Y., Dugan, J.B.: Approximate sensitivity analysis for acyclic Markov reliability models. IEEE Trans. Rel. **52**(2), 220–230 (2003)
47. Platzer, A., Quesel, J.D.: European train control system: A case study in formal verification. In: Proceedings of ICFEM, vol. 5885, pp. 246–265. Springer (2009)
48. Prescott, D., Andrews, J.: Modelling maintenance in railway infrastructure management. In: Proceedings of RAMS, pp. 1–6. IEEE (2013)
49. Rama, D., Andrews, J.D.: A reliability analysis of railway switches. Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit **227**(4), 344–363 (2013)
50. Rauzy, A.: New algorithms for fault trees analysis. Reliab. Eng. Syst. Saf. **40**(3), 203–211 (1993)
51. Ruijters, E., Budde, C.E., Nakhaee, M.C., Stoelinga, M., Bucur, D., Hiemstra, D., Schivo, S.: FFORT: a benchmark suite for fault tree analysis. In: Proceedings of ESREL, pp. 878–885. Research Publishing Services (2019)
52. Ruijters, E., Guck, D., van Noort, M., Stoelinga, M.: Reliability-centered maintenance of the electrically insulated railway joint via fault tree analysis: a practical experience report. In: Proceedings of DSN. IEEE (2016)
53. Ruijters, E., Stoelinga, M.: Fault tree analysis: a survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. **15–16**, 29–62 (2015)
54. Stamatelatos, M., Vesely, W., Dugan, J.B., Fragola, J., Minarick, J., Railsback, J.: Fault Tree Handbook with Aerospace Applications. NASA Headquarters (2002)
55. Stenström, C., Parida, A., Kumar, U.: Measuring and monitoring operational availability of rail infrastructure. Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit **230**(5), 1457–1468 (2016)
56. ter Beek, M.H., Borälv, A., Fantechi, A., Ferrari, A., Gnesi, S., Löfving, C., Mazzanti, F.: Adopting formal methods in an industrial setting: the railways case. In: Procedings of FM, LNCS, vol. 11800, pp. 762–772. Springer (2019)
57. Vanit-Anunchai, S.: Modelling railway interlocking tables using coloured Petri nets. In: Proceedings of COORDINATION, LNCS, vol. 6116, pp. 137–151. Springer (2010)
58. Vesely, W., Davis, T., Denning, R., Saltos, N.: Measures of risk importance and their applications. Technical report, Battelle Columbus Labs (1983)
59. Volk, M., Junges, S., Katoen, J.P.: Fast dynamic fault tree analysis by model checking techniques. IEEE Trans. Ind. Inf. **14**(1), 370–379 (2018)
60. Volk, M., Weik, N., Katoen, J.P., Nießen, N.: A DFT modeling approach for infrastructure reliability analysis of railway station areas. In: Proceedings of FMICS, LNCS, vol. 11687, pp. 40–58. Springer (2019)
61. Weik, N.: Long-term capacity planning of railway infrastructure: a stochastic approach capturing infrastructure unavailability. Ph.D. thesis, RWTH Aachen University (2020)
62. Weik, N., Nießen, N.: A quasi-birth-and-death process approach for integrated capacity and reliability modeling of railway systems. J. Rail Transp. Plan. Manag. **7**(3), 114–126 (2017)
63. Winter, K., Robinson, N.J.: Modelling large railway interlockings and model checking small ones. In: Proceedings of ACSC, pp. 309–316. Australian Computer Society (2003)