MISSION

Deliverable D2.1

Training: Fault Trees from Data and RES

Deliverable

Number and title:	D2.1 – Training: Fault Trees from Data and RES		
Work package:	WP2 (Reliability and Resilience)		
Lead author:	Matthias Volk (UT)		
Contributors:	Marielle Stoelinga (UT), Lisandro A. Jimenez-Roa (UT)		
Reviewers:	Thomas Noll (RWTH), Pedro R. D'Argenio (UNC)		
Due date (GA):	M6 (2022-03-31)	*Dissemination level:*	Public
Due date (revised):	2022-07-31	*Version:*	1.0 (final)
Submission date:	2022-07-08	*Pages:*	14

Version history

Version	*Date*	*Notes*
1.0	2022-07-08	First official release

Project

Title:	Models in Space Systems: Integration, Operation, and Networking		
Acronym:	MISSION	*Start date:*	01-10-2021
GA no.:	101008233	*Duration:*	48 months
Call:	H2020-MSCA-RISE-2020	*Website:*	mission-project.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

This document reflects only the authors' view. It does not represent the opinion of the European Union, and the European Union is not responsible for any use that may be made of the information it contains.

1 Introduction

Designing reliable systems requires formal models which accurately reflect reality as well as efficient analysis methods to assess the reliability of the modelled system. Both the modelling and the analysis come with specific challenges such as (1) how to efficiently create formal models and (2) how to efficiently perform analysis in the presence of rare events?

Creating formal models. Formal models are traditionally created by domain experts in a manual and error-prone manner. It is therefore desirable to generate such models in a (semi-)automated manner from either design documents or sensor data obtained during system operation. Automatic generation of formal models allows to speed-up the design process and in particular quickly adapt to changes by simply regenerating the model. In this document, we focus on the generation of fault tree models. Fault trees [73] are a common model in reliability engineering and widely used in industry. Fault trees model how failures occurring in the leaves, representing atomic components, propagate through a system and lead to a system failure. In Sect. 2, we summarize the current state-of-the-art for generating fault trees in a (semi-)automated manner.

Analysis in the presence of rare events. A common approach to assess the reliability of systems is to perform Monte Carlo simulation. Monte Carlo simulation can be applied to a broad range of systems and requires only a small memory footprint. However, the number of simulation runs required to obtain results with a desired confidence can be very large, in particular if rare events are present. Rare events occur only with very small probability and thus, render Monte Carlo simulation very costly. In Sect. 3, we summarize approaches for rare event simulation which are designed to require less simulation runs than standard approaches in the presence of rare events.

2 Automatic inference of fault tree models: an overview

The automatic *inference* (also referred to in the literature as *construction* [75], *synthesis* [25], or *induction* [48]) of fault tree models corresponds to the task of uncovering the structure (also called *skeleton* [45]) of a fault tree (FT) from a compatible source of information. The topic of automatic inference of FT models has been discussed since the 70's, about a decade after the birth of FTs.

We identified three categories of computer-assisted inference approaches, namely: (1) *knowledge-based*, (2) *model-based*, and (3) *data-driven* approaches. The main differences between these categories are that *knowledge-based* approaches employ different heuristics for knowledge representation, using information about the basic components and their interconnections in the system under analysis [38]. *Model-based* approaches translate existing system and/or graph models into FTs. *Data-driven* methods have structured databases with information on the status of the system and its components as primary source of information.

In the following, we provide an overview of the first two categories, and a detailed summary of the third category comparing the advantages and disadvantages of the different methods.

2.1 Knowledge-based approaches

The very first attempts to automate the generation of FTs fall into the category of *knowledge-based* approaches.

Table 1 enlists relevant references that fall in this category, accompanied by their publication year, and relevant aspects of their methodology. Several of these references were thoroughly reviewed in [14], where the authors identified *formalization* as the main concern regarding the automatic construction of FT models. They also reviewed what they called "artificial intelligence inspired approaches" corre-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

Table 1: Knowledge-based approaches for the automatic inference of FT models.

Reference(s)	*Year*	*Methodology*
Fussell [20]	1972	Synthetic Tree Model + Failure Transfer Functions
Powers and Tompkins Jr [63]	1973	System description + Mini FTs
Salem et al. [75]	1976	Decision Tables + System topological description + Backtracking
Taylor [78]	1982	Decision Tables + Mini FTs + P&ID
Poucet [62]	1983	Macro Component Models (MCM) + Transfer Logic Models (TLM) + Human knowledge
Kelly and Lees [35]	1986	Decision Tables + Mini FTs + Manual Feedback
Wang and Liu [84]	1993	Decision Tables + Virtual Transfer Component + Pruning procedures
Xie et al. [89]	1993	Knowledge Tree + P&ID
Hunt et al. [25]	1993	P&ID + Mini FTs + Propagation functions
Elliott [19]	1994	Graphical Function Block Diagram + Pruning techniques
Henry and Andrews [23]	1997	Digraph and Decision Tables methods
Szabó and Tarnai [77]	2000	Based on functional hardware and software descriptions
Papadopoulos et al. [61]	2001	Failure Functional Analysis + Failure Mode and Effect Analysis
Majdara and Wakabayashi [52, 53]	2009	Component based approach: Function tables and/or State Transition Table (similar to decision tables)

sponding to the early attempts based on *expert systems*. The latter are reviewed in [38]. From Table 1, we observe that each method makes use of different sorts of heuristics, system descriptions and human expertise, but most approaches have in common the use of the following ingredients:

- *Decision tables* are used to model the failure/function of components. The decision tables represent the causes of each output event in terms of their inputs.

- *Backtracking* enables the identification of the inputs for primary and underdeveloped events. Backtracking has considerable difficulty with loops and recursion [65].

- *Mini FTs* consist of an input event, a set of component conditions, and a set of output and state-change events [78].

- *Piping and Instrumentation Diagrams (P&ID)* are detailed diagrams that shows the piping and process equipment together with the instrumentation and control devices.

2.2 Model-based approaches

The *model-based* approaches fundamentally make use of existing models that describe the *behaviour* of the system of interest. FT models are then obtained by applying translation rules.

Table 2 summarizes some references of the approaches that fall in this category, accompanied by the model source. One of the biggest drawbacks of these approaches is the need of a pre-existing model [18].

Table 2: Model-based approaches for the automatic inference of FT models.

Reference(s)	*Year*	*Model source*
Lapp and Powers [37]	1977	Digraphs
De Vries [17]	1990	Digraphs
Andrews and Brennan [1]	1990	Digraphs
Liggesmeyer and Rothfelder [44]	1998	Finite state machines
Wang et al. [86]	2002	System block diagram
Joshi et al. [32]	2007	AADL models
Hussain and Eschbach [26]	2010	Sequence-based specification (SBS) + Model testing
Majdara and Wakabayashi [51]	2010	Component-based method + Function Table + State Transition Table + Trace-back algorithm
Xiang et al. [87]	2011	SysML models
Mahmud and Mian [50]	2013	AADL models
Mhenni et al. [54]	2014	SysML models
Herbert and Sharp [24]	2014	Model checking
Möhrle et al. [55]	2015	Component Fault Trees
Zhang et al. [90]	2015	Go models
Iyenghar et al. [27]	2016	Simulink models
Bozzano et al. [6]	2017	AADL models
Dickerson et al. [18]	2018	UML activity model
Wang et al. [85]	2018	Kripke structure + Feature-Labelled Transition System (FLTS)
Linard et al. [46]	2019	Bayesian Networks

2.3 Data-driven approaches

The third category corresponds to data-driven approaches. The main difference of these approaches compared to the previous ones is that structured data on the status of the system and its components must be part of the algorithm input. Data-driven approaches aim to build an FT model that best reflects the information contained in such a data set. Applications of machine learning techniques and data analytics fall into this category.

Table 3 summarizes relevant literature in data-driven methods for the automatic inference of FT models. For each algorithm, we give the corresponding reference, the year, and the name of the approach with a hyperlink to the respective online repository containing the implementation (if it exists). We shortly list some benefits and limitations per method.

To the best of our knowledge, the very first attempt to tackle the challenge in a data-driven manner was made by Madden and Nolan [48] with their *IFT* algorithm, which is based on Quinlan's ID3 algorithm to induce *Decision Trees (DTs)* [64]. The authors also continued with this work in the subsequent years [47, 49].

In the same direction, Berikov [5] further discussed the application of DTs in combination with multi-dimensional time series to infer FTs.

Mukherjee and Chakraborty [57] addressed this challenge via *linguistic analysis* and domain knowledge to identify the nature of the failure from short plain text descriptions of equipment faults. From this information, they generated a FT model.

Roth et al. [69] proposed a method that follows the *Structural Complexity Management (StCM)* methodology, combining the concepts of *Design Structure Matrix (DSM)* and *Domain Mapping Matrix (DMM)* into a *Multiple Domain Matrix (MDM)*, where their main goal is to deduce dependencies that later on

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

Table 3: Data-driven approaches for the automatic inference of FT models.

Reference	*Year*	*Algorithm*
Madden and Nolan [48]	1994	IFT
Madden [47]	1998	Hierarchical induction + DE/IFT
Madden and Nolan [49]	1999	DE/IFT
Berikov [5]	2004	Multidimensional time series + DT
Mukherjee and Chakraborty [57]	2007	Linguistic analysis + domain knowledge
Roth et al. [69]	2015	Following StCM
Nauta et al. [58]	2018	LIFT
Waghen and Ouali [82]	2019	ILTA
Linard et al. [45]	2019	FT-EA
Lazarova-Molnar et al. [39]	2020	DDFTA
Linard et al. [46]	2020	FT-BN
Waghen and Ouali [83]	2021	MILTA
Jimenez-Roa et al. [30]	2022	FT-MOEA
Jimenez-Roa et al. [31]	2022	SymLearn + FT-MOEA

are used to infer the Boolean logic operators of the FT models.

Inspired by *Causal Decision Trees* [43], Nauta et al. [58] proposed LIFT, an approach based on the *Mantel-Haenszel* statistical test.

Based on *Knowledge Discovery in Data set*, Waghen and Ouali [82] propose a method for hierarchical causality analysis called *Interpretable Logic Tree Analysis* (ILTA), that looks for patterns in a data set which are translated into *Interpretable Logic Trees*. In [83], they further extended their work to the *Multi-level Interpretable Logic Tree Analysis* (MILTA), which tackles the problem of multiple cause-and-effect sequences (the latter a limitation of their previous version, the ILTA algorithm) by incorporating Bayesian probability rules.

Linard et al. [46] proposed a method that consists of learning a *Bayesian Network* graph, which is later translated into a FT model.

Lazarova-Molnar et al. [39] presented an approach for *data-driven fault tree analysis (DDFTA)* based on time series of fault data, binarization techniques, minimal cut sets (MCS) and Boolean algebra. This algorithm also manages to infer FT models with a k-out-of-N voting gates.

Evolutionary algorithms. The first attempt based on *evolutionary algorithms* was carried out by Linard et al. [45]. The authors created an algorithm to generate FT models from a labelled binary fault data set based on a uni-dimensional cost function.

Similarly, Jimenez-Roa et al. [30] used multi-objective evolutionary algorithms to generate FT models from data. Their approach called *FT-MOEA* allowed to optimize for both the accuracy of the FT reflecting the input data as well as for a minimal size of the FT. The evaluation showed that the resulting FTs are significantly smaller than the FTs obtained by [45].

The FT-MOEA approach was recently extended by Jimenez-Roa et al. [31]. Their approach exploited symmetries and independent parts which can be automatically derived from the input data. The evaluation showed that this approach called *SymLearn* is significantly faster than FT-MOEA for fault trees where symmetries and/or independent modules are present.

3 Rare event simulation: an overview

In the presences of rare events, i.e., events which very seldom occur, standard Monte Carlo simulation approaches require a large amount of samples to make statistically justified error bounds. *Rare event simulation (RES)* [34, 71] overcomes this issue by employing techniques that reduce the number of samples necessary to gain statistically justified bounds.

Several books provide introductions to rare event simulation, e.g., [7, 33, 70, 13]. In particular, Rubino and Tuffin [70] provide an overview on existing techniques and present the application of RES in several application areas such as queueing systems or dependability analysis.

In RES, two prominent techniques exist to make the occurrence of rare events more like during simulation: *importance sampling* and *importance splitting*. We provide an overview of both techniques in the following.

3.1 Importance sampling

One approach for RES is *importance sampling* [22, 59]. The idea is to adjust the probabilities of rare events such that they become more likely. The simulation then requires less samples to gain statistically significant results. In the end, the analysis results are adjusted again to the original model to revert the effect of the previous bias.

Example. Consider the following example first presented in [10]. A (biased) coin yields heads with probability $\frac{1}{80}$, i.e., heads is a rare event. Instead of using this original coin, one can adjust the probabilities such that heads occurs with probability $\frac{1}{8}$ instead. Sampling the new coin 1000 times might yield heads in 103 instances. As obtaining heads is now 10-times more likely, each occurrence of heads is only counted as $\frac{1}{10}$ instead of 1. Thus, the resulting probability of the original coin is $\frac{103}{1000} \cdot \frac{1}{10} = 0.0103$.
The challenge in importance sampling is to adapt the probabilities in a “right” way. Events bringing the model closer to the desired state should be made more likely. However, if the “wrong” events are emphasized, e.g., events which are irrelevant, it might lead to worse performance than the original simulation. Finding a good heuristic how to bias the probability distributions is therefore crucial to obtain good performance of the simulation.

Standard importance sampling does not work well when the underlying distribution is heavy-tailed, e.g., following a Weibull or log-normal distribution. Asmussen et al. [2] presented two algorithms for heavy-tailed distributions, in particular for the event in which the sum of independent and identically distributed positive random variables must exceed a certain threshold.

3.2 Importance splitting

Another approach to perform rare event simulation is *importance splitting* [34, 3]. Importance splitting can be applied when rare events occur due to a sequence of (less rare) intermediate events. During simulation, sample paths can show a promising direction, i.e., intermediate events occurred which could lead to a rare event. Such partial paths are cloned and multiple simulation runs are started using the same partial path as prefix. That way, promising paths are given more weight during the simulation.

Example. Consider the following example from [10]. A coin is flipped 8 times in a row and the goal is to obtain heads at least 3 times. If heads is the outcome for the first coin toss, this partial path (H) is promising as only two further heads are required. The partial path (H) is cloned, for example 7 times. For each of the 7 paths starting with (H), subsequent coin tosses are performed independently. If in the end, one of the 7 simulation paths resulted in 3 times heads, this outcome is counted as $\frac{1}{7}$ instead of 1 due to the split. Splitting can also be performed multiple times. For example, the path (HH) resulting

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

from the previous path (H) can be split again, for example into 5 copies. As a result, observations resulting from (HH) are then counted as $\frac{1}{7} \cdot \frac{1}{5} = \frac{1}{35}$.

An important aspect of importance splitting is the decision when to split, i.e., which paths are promising? The algorithm uses an *importance function* to guide the splitting. The performance of the importance splitting heavily depends on the quality of the *importance function* [21, 71, 29].

RESTART One prominent algorithm for importance splitting is the *RESTART method (REpetitive Simulation Trials After Reaching Threshold)* by Villen-Altamirano et al. [81]. Originally, it was introduced to generalize previous approaches of importance splitting to most simulation models. RESTART operates similar to the original importance splitting. The main difference is the behaviour when simulation runs started in a split reach an importance which is lower than the importance in the split state where they were started, i.e., they become “less promising”. In RESTART, all but one of these simulation runs are terminated whereas importance splitting continues simulation for all runs. In addition, if such an “unpromising” run exceeds the importance threshold again, RESTART allows to perform new splits whereas importance sampling does not.

Villén-Altamirano [80] compared both RESTART and importance splitting. The conclusion is that RESTART behaves significantly better than importance splitting for the considered cases. The performance of RESTART can further be improved on models with many thresholds by using so called *prolonged trials*, where simulation runs are only terminated if they fall below previous thresholds (and not the one where they were cloned from).

In [79], the RESTART method is applied to many models of highly dependable systems. Their result showed that for balanced systems, i.e., redundant components which have the same probability of failure, the steady-state unavailability could be accurately estimated and only required short computation times. For unbalanced systems, the computations required significantly more time.

In [40, 41] importance splitting and in particular multi-level splitting is presented. The goal is to compute the probability of reaching a state B (which is rarely reached) before reaching (or returning to) a state A.

In previous approaches, the levels at which sample paths are split are fixed. In contrast, Cérou and Guyader [16] make use of adaptive levels which are computed on-the-fly. The authors prove that the estimation via adaptive levels is asymptotically consistent and requires only slightly more computation effort than the classical multilevel splitting.

3.3 Application domains

Rare event simulation has been applied in various domains. Rubino and Tuffin [70] provide an overview of RES in different application areas including queuing systems, counting problems, dependability analysis, particle transport, systems biology and a stochastic hybrid systems modelling air traffic.

In the setting of statistical model checking (SMC), RES is used to improve the performance. Jégourel et al. [28] use importance splitting for statistical model checking of rare properties. In particular, they derive importance functions based on the logical properties. Tool support for SMC is given for example by Plasma Lab [42], which incorporates both importance sampling and importance splitting. However, the importance function has to be manually given by the user. In contrast, modes [11] enables fully automated importance splitting without the need for user input. It combines RES with lightweight scheduler sampling to handle non-deterministic models.

O’Kelly et al. [60] applied RES to test an entire modern autonomous driving system. The authors used adaptive importance-sampling methods to estimate the probability of an accident for autonomous vehicles.

Ridder [68] applied importance sampling to estimate rare event probabilities in Markovian reliability systems. The importance sampling uses the cross-entropy method [72]. The approach is evaluated

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

Table 4: Rare event simulation approaches for DFT analysis.

Reference	*Year*	*Algorithm*	*Tool*
Ruijters et al. [74]	2019	Importance sampling with Path-ZVA + compositional state space generation	DFTRES
Budde et al. [10]	2020	Importance splitting	using FIG tool
Budde and Stoelinga [9]	2020	Importance splitting with importance function based on MCS	using FIG tool

on three discrete-time Markov chains modelling generic technology systems with various components which can fail and can be repaired.

Xiao et al. [88] performed rare event simulation on a model of a repairable system consisting of consecutive k-out-of-n components and non-exponential repair distributions. They applied three different techniques: importance sampling, conditional expectation estimation and a combination of the two. The evaluation showed that the combined method performs best for estimating the unreliability and unavailability. Conditional expectation estimation worked best for estimating the mean-time-to-failure (MTTF) and mean-time-between-failures (MTBF).

The method of Cérou et al. [15] considers rare events for fixed probability distributions. The approach uses a system of interacting particles and exploits a Feynman-Kac representation of that system to analyze their fluctuations. The authors show the relevance of the new algorithm in two application areas: watermarking and finger-printing of digital content.

Ramakrishnan [66] performed Monte Carlo simulation with importance sampling to analyse the unavailability and mean-time-to-failure of a shutdown system of a fast breeder nuclear reactor. The evaluation showed that a balanced failure biasing scheme in the importance sampling results in a better variance reduction compared to a simple failure biasing or standard Monte Carlo simulation.

Kumamoto et al. [36] applied Monte Carlo simulation to fault trees analysis. The authors calculated the system unavailability of a large complicated system. The approach is based on so-called *dagger-sampling* which requires a smaller number of uniform random numbers than standard Monte Carlo simulation.

3.4 Dynamic fault trees

Rare event simulation has been successfully applied for reliability analysis of systems [22, 71, 66, 36]. Recently, RES has become a focus of research for *dynamic fault tree (DFT)* analysis. Dynamic fault trees (DFT) [4, 73] are a dynamic extension of classical fault trees. The leaves in fault trees – both classical and dynamic ones – are typically rare events because failures should in the best-case be unlikely. Analysis of classical fault trees is usually performed via binary-decision diagrams [73] in which rare events are not an issue. However, dynamic fault trees require other analysis approaches, see [73]. One possibility is the analysis via Monte Carlo simulation [73] which requires an explicit handling of the rare events though. Approaches based on both importance sampling and importance splitting have been developed to automatically analyse DFTs. We list current works on DFT analysis via RES in Table 4 and summarize them in the following.

Importance sampling Ruijters et al. [74] present rare event simulation for dynamic fault trees with repairs. The method combines importance sampling with a compositional state space generation. The importance sampling is based on the Path-ZVA algorithm [67]. The evaluation of the tool FTRES showed that it outperforms the tool DFTCalc (based on model checking) for larger models and yields more accurate results than performing standard Monte Carlo simulation with the same time budget. Extending FTRES, the tool DFTRES [12] provides DFT analysis via RES.

Importance splitting Budde et al. [10] present rare event simulation for repairable DFT via importance splitting. While the importance function must commonly be given by domain or RES experts, the presented approach computes the importance function fully automatically. In addition, the approach handles both Markovian and non-Markovian probability distributions [56]. The tool-chain makes use of the FIG tool [8] for rare event simulation.

Following previous work, Budde and Stoelinga [9] present an approach in which the importance function is computed based on the tree structure. The tree structure is represented by the minimal cut sets. The importance is then given by the amount of failed elements in the cut sets.

Rare event approximation Lastly, a similarly named but different concept for fault tree analysis is *rare event approximation* [76]. This approach is not based on simulation but on minimal cut sets. For fault trees with shared events, the sum of the probabilities over all minimal cut sets is an approximation of the system failure probability. This approximation however is only accurate when the failures of shared events are improbable, i.e., rare.

4 Conclusion

We summarized existing works on automatic inference of fault tree models and rare event simulation.

Automatic inference of fault trees. Automatic inference of fault tree models can be categorized into knowledge-based, model-based and data-driven approaches. In recent years, the latter has gained attention and a number of approaches have been proposed to automatically generate fault tree models from data. However, all current approaches are still limited to small fault tree sizes and do not scale yet. Future research will therefore need to improve the scalability of the approaches. In addition, inference of fault trees also has challenges when it comes to rare events. Rare events might not be present in the data and therefore these events and their effects are not known. Future inference approaches need to take unknown events into account when constructing fault trees from data.

Rare event simulation. For rare event simulation of systems two main lines of approaches exist: importance sampling and importance splitting. Both approaches require domain knowledge to guide the biasing in importance sampling and to define the importance function in importance splitting, respectively. Dedicated functions for biasing and splitting have been developed for different domains. Recently, rare event simulation has been successfully applied to the reliability model of dynamic fault trees. Both importance sampling and importance splitting seem promising to handle large, realistic fault trees.

References

[1] John Andrews and Gerry Brennan. Application of the digraph method of fault tree construction to a complex control configuration. *RESS*, 28(3):357–384, 1990.

[2] Søren Asmussen, Klemens Binswanger, and Bjarne Højgaard. Rare Events Simulation for Heavy-Tailed Distributions. *Bernoulli*, 6(2):303–322, 2000. ISSN 1350-7265. doi: 10.2307/3318578.

[3] A. J. Bayes. Statistical techniques for simulation models. *Aust. Comput. J.*, 2(4):180–184, 1970.

[4] J. Bechta Dugan, Salvatore J. Bavuso, and M.A. Boyd. Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability*, 41(3):363–377, 1992. ISSN 0018-9529. doi: 10.1109/24.159800.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

[5] VB Berikov. Fault tree construction on the basis of multivariate time series analysis. In *Proceedings. The 8th Russian-Korean International Symposium on Science and Technology, 2004. KORUS 2004.*, volume 2, pages 103–106. IEEE, 2004.

[6] Marco Bozzano, Harold Bruintjes, Alessandro Cimatti, Joost-Pieter Katoen, Thomas Noll, and Stefano Tonetta. *Formal Methods for Aerospace Systems*, pages 133–159. Springer Singapore, Singapore, 2017.

[7] James Antonio Bucklew. *Introduction to rare event simulation*, volume 5. Springer, 2004.

[8] Carlos E. Budde. FIG: the finite improbability generator. In *TACAS (1)*, volume 12078 of *Lecture Notes in Computer Science*, pages 483–491. Springer, 2020.

[9] Carlos E. Budde and Mariëlle Stoelinga. Automated rare event simulation for fault tree analysis via minimal cut sets. In *MMB*, volume 12040 of *Lecture Notes in Computer Science*, pages 259–277. Springer, 2020.

[10] Carlos E. Budde, Marco Biagi, Raúl E. Monti, Pedro R. D'Argenio, and Mariëlle Stoelinga. Rare event simulation for non-Markovian repairable fault trees. In *TACAS (1)*, volume 12078 of *Lecture Notes in Computer Science*, pages 463–482. Springer, 2020.

[11] Carlos E. Budde, Pedro R. D'Argenio, Arnd Hartmanns, and Sean Sedwards. An efficient statistical model checker for nondeterminism and rare events. *Int. J. Softw. Tools Technol. Transf.*, 22(6): 759–780, 2020.

[12] Carlos E. Budde, Enno Ruijters, and Mariëlle Stoelinga. The dynamic fault tree rare event simulator. In *QEST*, volume 12289 of *Lecture Notes in Computer Science*, pages 233–238. Springer, 2020.

[13] Amarjit Budhiraja and Paul Dupuis. *Analysis and approximation of rare events*, volume 94 of *Representations and Weak Convergence Methods. Series Prob. Theory and Stoch. Modelling*. Springer, 2019.

[14] A Carpignano and A Poucet. Computer assisted fault tree construction: a review of methods and concerns. *RESS*, 44(3):265–278, 1994.

[15] Frédéric Cérou, Pierre Del Moral, Teddy Furon, and Arnaud Guyader. Sequential Monte Carlo for rare event estimation. *Stat. Comput.*, 22(3):795–808, 2012.

[16] Frédéric Cérou and Arnaud Guyader. Adaptive Multilevel Splitting for Rare Event Analysis. *Stochastic Analysis and Applications*, 2007. doi: 10.1080/07362990601139628.

[17] Ronald C De Vries. An automated methodology for generating a fault tree. *IEEE Trans. Reliab.*, 39(1):76–86, 1990.

[18] Charles E. Dickerson, Rosmira Roslan, and Siyuan Ji. A formal transformation method for automated fault tree generation from a UML activity model. *IEEE Trans. Reliab.*, 67(3):1219–1236, 2018.

[19] Margaret S Elliott. Computer-assisted fault-tree construction using a knowledge-based approach. *IEEE Trans. on Reliab.*, 43(1):112–120, 1994.

[20] Jerry B Fussell. A formal methodology for fault tree construction. *Nuclear Science and engineering*, 52(4):421–432, 1973.

[21] Marnix J. J. Garvels, Jan-Kees C. W. van Ommeren, and Dirk P. Kroese. On the importance function in splitting simulation. *Eur. Trans. Telecommun.*, 13(4):363–371, 2002.

[22] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Model. Comput. Simul.*, 5(1):43–85, 1995.

[23] JJ Henry and JD Andrews. Computerized fault tree construction for a train braking system. *Quality and Reliability Engineering International*, 13(5):299–309, 1997.

[24] Luke Thomas Herbert and Robin Sharp. Workflow fault tree generation through model checking. In *22nd ESREL conference*, pages 2229–2236. CRC Press, 2014.

[25] A Hunt, BE Kelly, JS Mullhi, FP Lees, and AG Rushton. The propagation of faults in process plants: 6, overview of, and modelling for, fault tree synthesis. *RESS*, 39(2):173–194, 1993.

[26] Tanvir Hussain and Robert Eschbach. Automated fault tree generation and risk-based testing of networked automation systems. In *ETFA*, pages 1–8. IEEE, 2010.

[27] Padma Iyenghar, Stephan Wessels, Arne Noyer, Elke Pulvermüller, and Clemens Westerkamp. A novel approach towards model-driven reliability analysis of simulink models. In *ETFA*, pages 1–6. IEEE, 2016.

[28] Cyrille Jégourel, Axel Legay, and Sean Sedwards. Importance splitting for statistical model checking rare properties. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 576–591. Springer, 2013.

[29] Cyrille Jégourel, Axel Legay, Sean Sedwards, and Louis-Marie Traonouez. Distributed verification of rare properties using importance splitting observers. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, 72, 2015.

[30] Lisandro A. Jimenez-Roa, Tom Heskes, Tiedo Tinga, and Mariëlle Stoelinga. Automatic inference of fault tree models via multi-objective evolutionary algorithms. *CoRR*, abs/2204.03743, 2022.

[31] Lisandro A. Jimenez-Roa, Matthias Volk, and Mariëlle Stoelinga. Data-driven inference of fault tree models exploiting symmetry and modularization. In *SAFECOMP*, LNCS. Springer, 2022. to appear.

[32] Anjali Joshi, Steve Vestal, and Pam Binns. Automatic generation of static fault trees from AADL models. 2007.

[33] Sandeep Juneja and Perwez Shahabuddin. Chapter 11 rare-event simulation techniques: An introduction and recent advances. In *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, pages 291–350. North-Holland, 2006.

[34] Herman Kahn and Theodore E Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, 12:27–30, 1951.

[35] BE Kelly and FP Lees. The propagation of faults in process plants: 1. modelling of fault propagation. *Reliability Engineering*, 16(1):3–38, 1986.

[36] Hiromitsu Kumamoto, Kazuo Tanaka, Koichi Inoue, and Ernest J Henley. Dagger-sampling Monte Carlo for system unavailability evaluation. *IEEE Transactions on Reliability*, 29(2):122–125, 1980.

[37] Steven A Lapp and Gary J Powers. Computer-aided synthesis of fault-trees. *IEEE Trans. on Reliab.*, 26(1), 1977.

[38] G Latif-Shabgahi. Comparing selected knowledge-based fault tree construction tools. In *Proc. IASTED Int. Conf. Intell. Syst. Control*, 2002.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.

[39] Sanja Lazarova-Molnar, Parisa Niloofar, and Gabor Kevin Barta. Data-driven fault tree modeling for reliability assessment of cyber-physical systems. In *WSC*. IEEE, 2020.

[40] Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Splitting for rare-event simulation. In *WSC*, pages 137–148. IEEE Computer Society, 2006.

[41] Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Rare events, splitting, and quasi-Monte Carlo. *ACM Trans. Model. Comput. Simul.*, 17(2):9, 2007.

[42] Axel Legay, Sean Sedwards, and Louis-Marie Traonouez. Plasma lab: A modular statistical model checking platform. In *ISoLA (1)*, volume 9952 of *Lecture Notes in Computer Science*, pages 77–93, 2016.

[43] Jiuyong Li, Saisai Ma, Thuc Duy Le, Lin Liu, and Jixue Liu. Causal decision trees. *IEEE Trans. Knowl. Data Eng.*, 29(2):257–271, 2017.

[44] Peter Liggesmeyer and Martin Rothfelder. Improving system reliability with automatic fault tree generation. In *FTCS*, pages 90–99. IEEE Computer Society, 1998.

[45] Alexis Linard, Doina Bucur, and Mariëlle Stoelinga. Fault trees from data: Efficient learning with an evolutionary algorithm. In *SETTA*, volume 11951 of *LNCS*. Springer, 2019.

[46] Alexis Linard, Marcos L.P. Bueno, Doina Bucur, and Mariëlle Stoelinga. Induction of fault trees through Bayesian networks. In *ESREL*, pages 910–917. Research Publishing, 2019.

[47] Michael G Madden. Hierarchically structured inductive learning for fault diagnosis. *WIT Trans. on Information and Communication Technologies*, 20, 1998.

[48] Michael G Madden and Paul J Nolan. Generation of fault trees from simulated incipient fault case data. *WIT Trans. on Information and Communication Technologies*, 6, 1994.

[49] Michael GM Madden and Paul J Nolan. Monitoring and diagnosis of multiple incipient faults using fault tree induction. *IEE Proceedings-Control Theory and Applications*, 146(2):204–212, 1999.

[50] N Mahmud and Z Mian. Automatic generation of temporal fault trees from AADL models. *ESREL*, pages 2741–2749, 2013.

[51] A Majdara and T Wakabayashi. Computerized fault tree construction for improved reliability analysis. *WIT Trans. on Information and Communication Technologies*, 43:149–160, 2010.

[52] Aref Majdara and Toshio Wakabayashi. Component-based modeling of systems for automated fault tree generation. *Reliab. Eng. Syst. Saf.*, 94(6):1076–1086, 2009.

[53] Aref Majdara and Toshio Wakabayashi. A new approach for computer-aided fault tree generation. In *2009 3rd Annual IEEE Systems Conference*, pages 308–312. IEEE, 2009.

[54] Faïda Mhenni, Nga Nguyen, and Jean-Yves Choley. Automatic fault tree generation from SysML system models. In *AIM*, pages 715–720. IEEE, 2014.

[55] Felix Möhrle, Marc Zeller, Kai Höfig, Martin Rothfelder, and Peter Liggesmeyer. Automated compositional safety analysis using component fault trees. In *ISSRE Workshops*, pages 152–159. IEEE Computer Society, 2015.

[56] Raúl E. Monti, Carlos E. Budde, and Pedro R. D'Argenio. A compositional semantics for repairable fault trees with general distributions. In *LPAR*, volume 73 of *EPiC Series in Computing*, pages 354–372. EasyChair, 2020.

[57] Saikat Mukherjee and Amit Chakraborty. Automated fault tree generation: bridging reliability with text mining. In *RAMS*, pages 83–88. IEEE, 2007.

[58] Meike Nauta, Doina Bucur, and Mariëlle Stoelinga. LIFT: learning fault trees from observational data. In *QEST*, volume 11024 of *LNCS*, pages 306–322. Springer, 2018.

[59] Victor F. Nicola, Perwez Shahabuddin, and Marvin K. Nakayama. Techniques for fast simulation of models of highly dependable systems. *IEEE Trans. Reliab.*, 50(3):246–264, 2001.

[60] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C. Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *NeurIPS 2018*, pages 9849–9860, 2018.

[61] Yiannis Papadopoulos, John Alexander McDermid, Ralph Sasse, and Gunter Heiner. Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure. *Reliab. Eng. Syst. Saf.*, 71(3):229–247, 2001.

[62] André Poucet. Computer aided fault tree synthesis. Technical report, Commission of the European Communities, 1983.

[63] Gary J Powers and Frederick C Tompkins Jr. Fault tree synthesis for chemical processes. *AIChE Journal*, 20(2):376–387, 1974.

[64] J. Ross Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.

[65] Andrew Rae and Peter Lindsay. A behaviour-based method for fault tree generation. In *Int. System Safety Conference, System Safety Society*, pages 289–298, 2004.

[66] M Ramakrishnan. Unavailability estimation of shutdown system of a fast reactor by Monte Carlo simulation. *Annals of nuclear energy*, 90:264–274, 2016.

[67] Daniël Reijsbergen, Pieter-Tjerk de Boer, Werner R. W. Scheinhardt, and Sandeep Juneja. Path-zva: General, efficient, and automated importance sampling for highly reliable Markovian systems. *ACM Trans. Model. Comput. Simul.*, 28(3):22:1–22:25, 2018.

[68] Ad Ridder. Importance sampling simulations of Markovian reliability systems using cross-entropy. *Ann. Oper. Res.*, 134(1):119–136, 2005.

[69] Michael Roth, Moritz Wolf, and Udo Lindemann. Integrated matrix-based fault tree generation and evaluation. *Procedia Computer Science*, 44:599–608, 2015.

[70] Gerardo Rubino and Bruno Tuffin, editors. *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.

[71] Gerardo Rubino and Bruno Tuffin. Introduction to rare event simulation. In *Rare Event Simulation using Monte Carlo Methods*, pages 1–13. Wiley, 2009.

[72] Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

[73] Enno Ruijters and Mariëlle Stoelinga. Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Comput. Sci. Rev.*, 15:29–62, 2015.

[74] Enno Ruijters, Daniël Reijsbergen, Pieter-Tjerk de Boer, and Mariëlle Stoelinga. Rare event simulation for dynamic fault trees. *Reliab. Eng. Syst. Saf.*, 186:220–231, 2019.

[75] Steven L Salem, GE Apostolakis, and David Okrent. Computer-oriented approach to fault-tree construction. Technical report, California Univ., 1976.

[76] Michael Stamatelatos, William Vesely, Joanne Dugan, Joseph Fragola, Joseph Minarick, and Jan Railsback. Fault tree handbook with aerospace applications. 2002.

[77] Géza Szabó and G Tarnai. Automatic fault-tree generation as a support for safety studies of railway interlocking systems. *IFAC Proceedings Volumes*, 33(9):437–442, 2000.

[78] JR Taylor. An algorithm for fault-tree construction. *IEEE Trans. on Reliab.*, 31(2):137–146, 1982.

[79] José Villén-Altamirano. Importance functions for RESTART simulation of highly-dependable systems. *Simul.*, 83(12):821–828, 2007.

[80] José Villén-Altamirano. RESTART vs splitting: A comparative study. *Perform. Evaluation*, 121-122: 38–47, 2018.

[81] Manuel Villen-Altamirano, Jose Villen-Altamirano, et al. Restart: a method for accelerating rare event simulations. *Queueing, Performance and Control in ATM (ITC-13)*, pages 71–76, 1991.

[82] Kerelous Waghen and Mohamed-Salah Ouali. Interpretable logic tree analysis: A data-driven fault tree methodology for causality analysis. *Expert Syst. Appl.*, 136:376–391, 2019.

[83] Kerelous Waghen and Mohamed-Salah Ouali. Multi-level interpretable logic tree analysis: A data-driven approach for hierarchical causality analysis. *Expert Syst. Appl.*, 178:115035, 2021.

[84] JD Wang and TS Liu. A component behavioural model for automatic fault tree construction. *RESS*, 42(1):87–100, 1993.

[85] Lisong Wang, Sijie Li, Ou Wei, Mingyu Huang, and Jun Hu. An automated fault tree generation approach with fault configuration based on model checking. *IEEE Access*, 6:46900–46914, 2018.

[86] Y Wang, T Teague, H West, and S Mannan. A new algorithm for computer-aided fault tree synthesis. *Journal of Loss Prevention in the Process Industries*, 15(4):265–277, 2002.

[87] Jianwen Xiang, Kazuo Yanoo, Yoshiharu Maeno, and Kumiko Tadano. Automatic synthesis of static fault trees from system models. In *SSIRI*, pages 127–136. IEEE Computer Society, 2011.

[88] Gang Xiao, Zhizhong Li, and Ting Li. Dependability estimation for non-Markov consecutive-k-out-of-n: F repairable systems by fast simulation. *Reliab. Eng. Syst. Saf.*, 92(3):293–299, 2007.

[89] Gang Xie, Dazhi Xue, and Shuren Xi. Tree-expert: a tree-based expert system for fault tree construction. *RESS*, 40(3):295–309, 1993.

[90] Yanhua Zhang, Yi Ren, Linlin Liu, and Zili Wang. A method of fault tree generation based on go model. In *ICRSE*, pages 1–5. IEEE, 2015.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233.