Safe and Secure Future AI-Driven Railway Technologies: Challenges for Formal Methods in Railway

Seisenberger, Monika; ter Beek, Maurice H.; Fan, Xiuyi; Ferrari, Alessio; Haxthausen, Anne E.; James, Phillip; Lawrence, Andrew; Luttik, Bas; van de Pol, Jaco; Wimmer, Simon

Published in:
Leveraging Applications of Formal Methods, Verification and Validation. Practice

Link to article, DOI:
10.1007/978-3-031-19762-8_20

Publication date:
2022

Document Version
Peer reviewed version

Safe and Secure Future AI-Driven Railway Technologies: Challenges for Formal Methods in Railway

Monika Seisenberger ✉ [1], Maurice H. ter Beek [2], Xiuyi Fan [3], Alessio Ferrari [2], Anne E.Haxthausen [4], Phillip James [1], Andrew Lawrence [5], Bas Luttik [6], Jaco van de Pol [7], and Simon Wimmer [7]

[1] Swansea University, UK
m.seisenberger@swansea.ac.uk
[2] ISTI–CNR, Pisa, Italy
[3] Nanyang Technological University, Singapore
[4] Technical University of Denmark, Denmark
[5] Siemens Mobility Chippenham, UK
[6] Eindhoven University of Technology, The Netherlands
[7] Aarhus University, Denmark

Abstract. In 2020, the EU launched its sustainable and smart mobility strategy, outlining how it plans to have a 90% reduction in transport emission by 2050. Central to achieving this goal will be the improvement of rail technology, with many new data-driven visionary systems being proposed. AI will be the enabling technology for many of those systems. However, safety and security guarantees will be key for wide-spread acceptance and uptake by Industry and Society. Therefore, suitable verification and validation techniques are needed.
In this article, we argue how formal methods research can contribute to the development of modern Railway systems — which may or may not make use of AI techniques — and present several research problems and techniques worth to be further considered.

1 Introduction

In 2020, the EU launched its sustainable and smart mobility strategy, outlining how it plans to have a 90% reduction in transport emission by 2050. This will be key to achieving the European Green deal of becoming carbon neutral by 2050[8]. Central for this reduction will be the improvement of rail technology, as rail is one of the greenest modes of transportation. To address this ambition and support the interoperability and efficiency in the rail domain, new visionary systems based on interdisciplinary approaches in Engineering and Computer Science are being proposed, for example, innovative signalling systems (including moving block technology), smart monitoring and maintenance, optimal scheduling, and automated driving. Underpinning these new systems, data is recognized as a highly valuable asset. For instance, the UK's Rail Technical Strategy (10/2020) states: "Data will have fit for purpose governance, access arrangements, systems and technical skills. These building blocks underpin the progression of all the other functional priorities which each have their own specific data requirements and opportunities."[9] AI will be the enabling technology for such data driven systems. However, safety and security guarantees will be key for wide-spread acceptance and uptake by industry and society. Therefore, suitable verification and validation techniques are needed.

In the Railway Industry and in verification and validation research, there is a skills gap in terms of knowledge of AI principles, techniques, and practices. The Railway Industry is normally focused on developing software and systems with fully predictable, explainable and verifiable behaviour, while AI techniques are by nature adaptive, and open to different scenarios. More importantly, AI-based systems have explainability problems that collide with the idea of fully controllable and verifiable system behaviour. Therefore, the Railway Industry needs to exploit AI systems to deliver smart and green transport, whilst at the same time maintaining the highest standards in terms of safety and certification.

In the following, we propose several research questions and challenges to the formal methods community, to aid the development as well as the certification of safe and secure next generation rail

[8] https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en
[9] https://railtechnicalstrategy.co.uk/data-driven/

systems, many of which will make of use of AI techniques. Currently, ERTMS/ETCS level 3[10] is in the process to replace the traditional discrete train separation mechanism (blocks protected by hardware installed along the tracks) by a continuous mechanism (moving block), and use radio communication for the exchange of position information between trains and track-side. Also, Automatic Train Operation (ATO) will, at the higher grades of automation, replace the human driver and require a continuous software-controlled interaction, not only of a train with the track-side system, but also of trains between each other. Future signalling systems will be an order of magnitude more complex than they are today, and formal modelling and verification technology will be essential to cope with that complexity. The three challenges presented in this article will relate directly or indirectly to this situation.

The first challenge concerns certification and the associated verification technology. Automation of the verification is required to address the massive scale and complexity of railway systems. This entails intricate symbolic and parallel verification algorithms. A major problem for certification is that the implementation of the verification tools themselves could contain errors. Needed is a theory to equip verification tools with certificate generation, thus enabling the answers of automated verification tools to be checked independently. Furthermore, there is a need to investigate how AI based systems can be certified as this is not in general possible on the basis of today's CENELEC standards [47, 46, 48].

The second challenge concerns the European Train Traffic Management System. ERTMS/ETCS level 3 is anticipated to be the main railway system in 10 years' time. By introducing true continuous (moving block) signalling, it allows for both capacity increases and lower energy consumption, with trains effectively and intelligently managed. However, it also introduces elements of a hybrid nature into sub-systems of ERTMS, with components working with both discrete- and continuous-valued data. Along with this, traditional modelling of both safety and security need to change to include this hybrid nature. ERTMS consists of many interconnected components that differ in nature. This provides modelling challenges in both terms of complete models of the system, but also in terms of understanding and modelling the safety and security requirements of the sub-systems, both individually and as a whole. In addition, such signalling projects are often large and involve various aspects of backward compatibility with older signalling systems (e.g., to deal with older rolling stock) which causes scalability issues for current techniques. Modelling and verifying hybrid systems of the size presented by ERTMS/ETCS level 3 is an open challenge for Computer Science. Needed is a complete model which includes the safety and security requirements it must uphold. This will also serve as a reference architecture for ERTMS/ETCS level 3 deployment projects and help with the faster roll-out.

The third challenge concerns the extension of formal methods to include the use of AI techniques. We will present in Section 4 how this can be achieved, and highlight some of the methods we plan to utilise, as well as the problems that need to be overcome. Our focus on AI integration aligns well with the planned successor of Shift2Rail, Europe's Rail Joint Undertaking (EU-Rail), which will specifically focus on digitalisation and automation. So far, there are very few research projects and white papers (cf., e.g., [24, 2]) that look into the integration of AI into the Railway domain. Notable exceptions are the RAILS project (Roadmaps for A.I. Integration in the RaiL Sector)[11], which provided a first overview on available AI techniques, as well as application areas and work done in the Railway domain, and the TAURO project (Technologies for Autonomous Rail Operation)[12]. We aim to complement this research by focusing on the specific challenges for formal methods, from a methodological point of view and by means of specific case studies.

Finally, to complete the picture, we mention a few recent projects which address the usage of formal methods in various Railway areas and whose results should be beneficial for the proposed challenges. ASTRail (SAtellite-based Signaling and Automation SysTems on Railways along with Formal Method and Moving Block Validation)[13] studied how to enhance the ERTMS with satellite-based GNSS train positioning, moving block distancing, and automatic train driving by exploiting cutting-edge technologies from the automotive and avionics domains as well as suitably assessed formal methods. 4SECURail (FORmal Methods and CSIRT (Computer Security Incident Response Team) for the Railway sector)[14] provided a demonstrator of state-of-the-art formal methods and tools with an evaluation of

[10] https://www.ertms.net/, https://www.era.europa.eu/activities/european-rail-traffic-management-system-ertms/
[11] https://cordis.europa.eu/project/id/881782
[12] https://cordis.europa.eu/project/id/101014984
[13] http://www.astrail.eu
[14] https://www.4securail.eu/

the cost/benefit ratio and learning curves for adopting the demonstrator in the railway environment. It also developed, tested, and validated a CSIRT model and prototype co-designed with the relevant rail stakeholders. Notable new projects on intelligent systems are IN2SMART2[15], which is concerned with smart maintenance of Railway assets, SMART2[16], which aims at an integrated automated system for obstacle and track intrusion detection, and PERFORMINGRAIL (PERformance-based Formal modelling and Optimal tRaffic Management for movING-block RAILway signalling)[17].

2 Certified Verification of Railway Designs

Railway signaling systems are complex and safety-critical, imposing high safety standards and strict certification requirements [51]. The need for extensive testing, verification, and certification imposes an unfortunate barrier to the quick adoption of innovative railway technologies, which are essential to provide interoperability between national systems and increase utilisation of the railways. Therefore, we propose to push automated verification techniques for their certification. The dynamic nature of moving blocks, communication between trains and infrastructure, and the need to anticipate malfunctioning hardware, lead to an explosion of interactions and case distinctions, which can hardly be managed manually. Consequently, to eliminate manual errors and provide the necessary scalability, the verification needs to be *automated*. Moreover, the verification process needs to interact with the overall railway engineering process, in particular certification by regulatory bodies. To cater for the needs of stakeholders in this process (e.g., non-verification engineers, regulators, auditors), verification needs to be *trustworthy* and *explainable* [59, 105].

In the following, we will discuss the state-of-the-art and open challenges of verification technology regarding railway systems, and on the three axis of automation, trustworthiness, and explainability. We will also address the challenge of which standards to use for certifying train control (sub-)systems, specifically when they are based on AI technology.

2.1 Automated Verification

Automated verification of railway signalling poses a scientific challenge for several reasons, given next.

Complexity: As discussed above, the modern railway systems lead to highly complex designs (of software, hardware, systems). Here, based on our expertise, we focus primarily on the area of model checking for the automated verification of such designs. Model checking is a purely automatic method, which decides if a given specification (model) satisfies a given requirement (property). Various model-checking algorithms, ranging from exhaustive (probabilistic) model checking [9] to statistical model checking [3], have been proposed. Model checking is based on discrete enumeration [37], while statistical model checking involves running a sufficient number of (probabilistic) simulations to obtain statistical evidence (with a predefined confidence level).

Parametricity: For an effective deployment of signalling systems, there is a need to verify standard components that can be combined and instantiated to particular situations (e.g., track layouts). However, parametric verification is in general an undecidable problem [6]. This means that some form of abstraction and manual intervention is necessary to achieve verification results of the required generality. Compositional verification techniques [80] can be employed to combine verified standard components.

Continuous reasoning: Obviously, railway systems need to operate in real-time! Moreover, innovations like moving blocks require reasoning about continuous variables, such as braking curves. Stochastic behaviour (like failure of hardware components) leads to other forms of continuous reasoning. These continuous extensions remain challenging, and sometimes even cross the border of computational decidability [65]. Moreover, in case of AI-based systems, some components might be machine-learned models. These are of an inherently probabilistic nature and can be considered to be black-box components around which one may need to build a safety shield for runtime enforcement of guarantees [94].

[15] https://cordis.europa.eu/project/id/881574
[16] https://cordis.europa.eu/project/id/881784
[17] https://cordis.europa.eu/project/id/101015416

Model checking is essentially a smart enumeration method of the possible system behaviour. Typically, the properties are checked on-the-fly, during the enumeration. The major threat to model checking is the state space explosion, exhausting not only time resources, but also the memory of the computer. Statistical model checking scales better, since there is no need to explore the full state space and the required simulations can trivially be distributed and run in parallel, but contrary to model checking, exact results (with 100% confidence) cannot be achieved.

The field of model-checking algorithms has advanced tremendously over the last 25 years. Many improvements are based on symbolic reasoning algorithms, like abstraction, symmetry reduction, partial-order reduction, and the use of BDD, SAT, and SMT solvers. Other improvements deploy high-performance computing, like clusters of machines, multi-core hardware, or even many-core GPUs to off-load intensive verification tasks [73]. While model checking is a very active field of research with many open challenges we refrain from addressing them here. Instead, we consider issues more specific to the railway domain in the following.

2.2 Trustworthy and Certifiable Verification

Verification can be seen as an instance of *trust reduction*: when employing verification in the engineering process, trust in the safety of the resulting system will to a large extent be deduced from trust in the correctness of the model. Therefore, on the one hand, the model needs to be a suitable representation of the system and its environment. On the other hand, the results of the verification process need to be trustworthy. We focus on the latter issue, which poses three fundamental challenges:

1. Model-checking algorithms and implementations became so advanced and complicated that it is hard to guarantee that these verification tools themselves are free from bugs. Indeed, several cases of bugs in verification tools (and even verification theory [29, 98]) have been reported in the literature. This is clearly a threat for the certification of safety-critical systems.
2. Model checkers often run in so-called "bug hunting" mode. This means that they are specialised in finding bugs. To save on computational resources (time, memory, energy), they use various search heuristics. These heuristics give up completeness of the search. Hence, not finding any bugs is no guarantee of correctness. In this sense, bug hunting with model checking is similar to advanced, automated testing, which is also inherently incomplete (as is statistical model checking).
3. Model checking provides an asymmetric method. In principle, the answer to a model-checking query is either yes or no (the model either satisfies the property or violates it). In the "no-case", the model checker typically returns a counterexample, which can be inspected, tested on the model or even on the real system, and used to debug the model or the property. However, in the "yes-case", no further evidence is provided[18]. This is very unfortunate for certification, not only in light of the previous limitations, but also since no information for the safety-case is provided, besides the fact that no bug has been found.

A simple way to ameliorate implementation errors (but not errors in the theory), would be to compare the outputs of multiple tools. To achieve highest levels of trustworthiness, one might proceed by formally verifying the model checker. This provides a rigorous mathematical proof, checked in an interactive theorem prover [96, 102] (like Coq [23] or Isabelle/HOL [87]). This approach would ensure that the "yes-result" of the model checker is trustworthy. However, although possible in principle, verification of an advanced model checker would be a major undertaking. There is some progress in the formal verification of model-checking algorithms [92, 88] and even code [45, 21, 26, 30, 104], but so far, this could only be applied to relatively simple algorithms and basic implementations. Consequently, the "verified model checkers" cannot match the efficiency of high-performance model checkers.

We propose *certified model checking*, which provides a sweet-spot here: In this approach, one uses high-performance, "unsafe" model checkers, but equips them with the potential to generate some form of certificates in the "yes-case". One builds a separate, independent certifier, which checks the certificates. Since checking certificates is much simpler than finding certificates, the certifier is a relatively simple tool, which can be formally verified. If the formally verified certifier accepts the certificate generated by the model checker, then we achieve maximal confidence in the safety of the system-under-study. Hence, in certified model checking we combine maximal efficiency (high-performance model checkers) with maximal confidence (formally verified certifiers). The concept of certified model

[18] Note that this holds for safety properties, but for others the cases can be inverted.

checking was conceived for the μ-calculus [86]. A proof of concept has been provided for SMT-based model checking [74] (where certificates are basically invariants) and more recently for liveness checking of finite-state systems [60] and timed automata [103] (where certificates consist of reachability invariants and topological ranking functions). In the latter case, the certifier was fully verified in the theorem prover Isabelle/HOL.

Certified model checking poses multiple challenges. The generated certificates should contain sufficient information to check the proof independently; yet be more concise than the full state space, and easier to check than to generate. Independent certificate checker need to be constructed, possibly based on an interactive theorem prover. The certificate checker should be simple and amenable for formal verification. This requires a formalisation of (part of) the meta-theory of the verification tools.

The certified model-checking method is in its infancy: It is an open question what certificates for complicated model-checking algorithms, like partial-order order or symmetry reduction, should look like. It is also unclear how to generate certificates from parallel implementations of on-the-fly model checkers. The formal verification of the certification theory and the certifiers remains a challenge for these novel applications.

Moreover, a *lack of standardization* in modelling formalisms makes it hard to define general-purpose model checking certificates and to provide checkers for them. This is also a particular concern regarding the verification of the certificate checkers, as a large part of the laborious formalisation process would need to be repeated for each modelling language. Naturally, a lack of standardization is equally challenging for collaboration with regulatory bodies.

Finally, the certified verification approach needs to be extended to AI-intensive systems. In particular, classifiers and schedules generated by game-based AI algorithms should be equipped with certificates, so their essential properties can be checked independently. Interestingly, several model checking techniques, originally designed to analyse systems, can be extended to synthesis tasks. Tools like Uppaal TIGA [19] and Uppaal Stratego [40] can solve real-timed games. PRISM-games [75] can solve stochastic games. The winning strategies generated by such algorithms can (in principle) be converted to safe (and optimal) controllers. The synthesis of safe and optimal driving strategies with Uppaal Stratego has recently been shown for ERTMS Level 3 moving block railway signalling [13]. However, certification of synthesis algorithms is still open. In particular, the generated controllers tend to be large and enumerate possibilities, rather than conditions on data. Recent work [8] proposed to use decision trees to represent winning strategies; this could be a useful approach to synthesise controllers that are not just correct and certifiable but also explainable.

2.3 Explainable Verification

While certifiable verification can significantly increase reliability of the verification process, it is a whole challenge in itself to transform the computed certificate into an understandable piece of evidence, which can contribute to the safety case for certification authorities. This explainable verification should be the ultimate goal of this line of research. Without aspiring to completeness, we identify some open challenges in explainable verification.

Documentation of verification. In particular, this should contain the precise claim of what has been verified by the tools, including the modelling assumptions, the assumptions on the environment, and potential approximations and inaccuracies implied by the selected options in the verification tools.

Interactive explainability. It could be fruitful to base explainable verification methods on counterfactual explanation techniques for AI [67]. Counterfactuals provide an understanding into AI models by identifying similar inputs with changes in decisive properties that lead to a different model outcome than the one under study. For instance, given a particular dangerous scenario, the verification tool should be able to generate an argument why this particular scenario cannot happen. Novel interactive approaches that allow in-depth investigation of these properties for verification models will be needed. This is important to increase the trust of domain experts and certification authorities in the verification technology.

Practicality. Application-oriented research is required to investigate if certified model checking can provide useful evidence for the safety-case of railway systems, so that it significantly speeds-up the (regulatory) certification process for novel railway technology. This requires a careful consideration of the current standards used for certification in railways.

2.4 Standardisation

Certification of autonomous train control systems with Grade of Automation GoA 4 (unattended train operation, neither the driver nor the staff are required) in open railway environments, is a challenge: while conventional train control sub-systems can be certified on the basis of today's CENELEC standards [47, 46, 48], this is not the case for all AI-based sub-systems. The certification of such AI-based systems will require extensions/modifications of the current CENELEC standards or additional use of other standards. Therefore, there is a need to investigate how that can be done. In [90], Peleska et al. have investigated how the ANSI/UL 4600 pre-standard for Evaluation of Autonomous Products [100] can be used as a supplement to the CENELEC standards to certify autonomous freight trains and metro trains based on AI technology.

3 Certified Verification of Railway Designs

Historically, the application of formal methods in order to verify railway systems is well established within academia and although several success stories of formal development and verification of software for the railway domain have shown the potential advantages [51], these technologies are still not universally part of the usual toolboxes of railway signalling companies.

As early as 1995, formal methods were applied to verify interlockings [63, 16, 61, 5]. Since then, and indeed recently, newer approaches to interlocking verification have also been proposed, also at ISoLA, and have been shown to scale well to modern industrial systems [22, 36, 44, 64, 52, 50, 106, 68, 69, 70, 79, 32, 66, 20, 27]. In spite of this, such approaches still lack widespread use within the Rail industry often due to questions surrounding the usability and expertise required for applying formal methods [54, 53].

Railway infrastructure managers have started to use semi-formal modelling languages (e.g., UML and SysML) to specify requirements, but they often still have to delegate formal verification activities to academic partners. The culprit is that effectively using state-of-the-art formal verification technology requires a thorough academic background in formal methods. A formal method that can be used by railway engineers needs to facilitate modelling railway systems concisely at the right level of abstraction and it should be easily parametrisable with relevant data (e.g., a track layout). Also, it must allow for a straightforward specification of relevant safety and security properties, verification algorithms that scale for systems in the railway context, and provide insightful presentation of verification results.

In addition, in the next two decades, European railway infrastructure managers need to sustain an enormous growth in mobility by increasing the capacity of their networks at acceptable costs. Key to the capacity increase will be the introduction of innovative digital systems such as ERTMS level 3, which involves a radically new approach to train separation, and various forms of ATO. The smooth roll-out of such systems on the dense European railway networks is an enormous challenge. It is, e.g., unacceptable for a railway line to be unavailable for long periods, and it should be possible for the innovative system to coexist with the legacy system. Therefore, it is important to thoroughly prepare roll-out of new systems. Extensive use of formal modelling and analysis techniques in the development process will reduce the need for testing in the field.

3.1 Domain Specific Technology and Usability

Railway infrastructure managers have started to use (semi-)formal languages to model their systems. These models are typically very detailed and use concrete data. For an effective formal analysis, a domain-specific modelling language that supports the appropriate level of abstraction is essential. Furthermore, the modelling language should offer a means to model relevant continuous aspects of railway systems (e.g., braking curves). Safety and liveness properties must be formulated at the same level of abstraction. The latter is non-trivial in the context of the railway domain since normally railway engineers are not used to formulating safety requirements at the appropriate level of abstraction. It will be necessary to develop a property language that is, on the one hand, expressive enough to express relevant safety requirements, and on the other hand can be used by railway engineers. Formal verification in the railway sector has focussed on safety properties. The verification of liveness properties has not received much attention but is highly relevant in view of dependability of railway systems. For the verification of liveness properties, one typically needs to incorporate progress or fairness assumptions

in the verification process. Verification technology for the domain-specific modelling language should be built on top of a state-of-the-art general-purpose verification engine.

Another concern that is limiting uptake is the need to model in such a way that requirements can be efficiently verified for all relevant track layouts. With virtual fixed or moving blocks, data-parameterised verification becomes more important because track layouts are dynamically configurable. Currently, formal verification by model checking must often be carried out for specific track layouts. There is a need to develop formal modelling and verification technology that can efficiently verify safety and liveness properties of a signalling system for a class of relevant track layouts. To this end, the modelling language should facilitate data parametrisation. Not all data are realistic and so any approach will need to investigate ways of efficiently expressing assumptions on the parametrisation domain (e.g., using probability distributions). These assumptions must then be considered in the verification activities. Finally, a core aspect of this verification process is that it must be usable by railway engineers, a non-trivial endeavour. For railway-specific modelling and verification technology it is a major concern that railway engineers without extensive formal methods expertise can use it to gain insights in their systems. To this end, the modelling language should be easy to use, and verification results should be visualised (e.g., by running a graphical simulation of a counterexample). Railway engineers verify their designs by considering how they behave with respect to various operational scenarios. Model-checking technology can be used to generate interesting operational scenarios as evidence for certain properties. The idea is to formulate meaningful properties to verify and obtain the operational scenarios as evidence.

3.2 Standardised Reference Architectures

To facilitate interoperability, European railway infrastructure managers and operators and railway supply industry pursue standardisation of command and control systems. The best-known example is ERTMS/ETCS, which aims to standardise train-trackside communication (GSM-R), the train control system (ETCS), and the train management layer (ERTMS). The standard is formulated mostly in natural language and is therefore inherently ambiguous, which hampers a smooth deployment. Another standardisation project is EULYNX[19]; it aims at standardising the interfaces between the components of signalling systems (interlockings and field elements such as points, level crossings, light signals, etc.). In EULYNX the official standard is still formulated in natural language, but there is also an explicit aim to supplement standard with SysML models. The academic project FormaSig[20] develops verification and model-based testing technology by which these SysML models can be formally verified and used for model-based testing purposes [28]. Similar to EULYNX, railway operators have started OCORA in which a standardised modular architecture for on-board command and control equipment is developed. This effort too should be supported with formal methods.

Although parts of the ETCS standard have been formally modelled and analysed [91, 34, 35, 25, 10, 11, 20, 12, 14], it would be both challenging and highly desirable to develop a formal model of ETCS level 3. The ongoing development by railway infrastructure managers of an RCA (the reference Control Command and Signalling (CCS) architecture), a common reference architecture for railway command and control systems will be a convenient vehicle. This reference architecture consists of standardised components (e.g., interlocking and field elements with EULYNX-compliant interfaces), which will facilitate the deployment of innovative systems. Here, one particular challenge that is still open is to integrate ETCS level 3 with RCA by developing an integral model including all relevant components, in order to analyse the correctness of the interactions between those components and determine if the safety and security requirements are met. Of course this also requires a systematic analysis of both safety and security requirements for ETCS level 3.

3.3 Digital Railway Innovations

A number of upcoming digital railway innovations bring promises in terms of improved safety, capacity and resilience. With these adaptations to infrastructure come fresh challenges for formal modelling and verification in particular throughout the certification process of these systems.

[19] https://eulynx.eu
[20] https://fsa.win.tue.nl/formasig

A significant increase in capacity can be realised by going from blocks protected by train detection equipment to train separation based on more precise position information from the train. The introduction of such a new train separation system entails new challenges for the signalling system. For instance, it needs to be robust against radio connection problems between train and trackside and take into account inaccuracies in positioning information. But most importantly, for a significant period of time such a new system needs to coexist together with the old system. To deal with such issues, digital solutions are developed, and these are an order of magnitude more complex. An example is the Hybrid ERTMS/ETCS Level 3 concept, which tries to bring the flexibility of moving block train separation to a signalling system based on traditional trackside train detection [58]. The application of formal modelling and analysis techniques have proved to be beneficial for improving the specification of the concept and for coping with its complexity [10, 31, 7, 38, 41, 62, 81, 57, 15].

A further increase in capacity, energy consumption and reliability is expected to come from ATO. Railway infrastructure managers and operators are currently experimenting with a form of semi-automatic train operation. The automatic train operation system merely assists the driver with accelerating and decelerating efficiently, but the driver remains responsible for safe movement of the train. The next step is to integrate a form of ATO with ERTMS/ETCS. This integration serves as the perfect example of a complex hybrid system and thus modelling and verification challenges that exist for hybrid systems apply. In particular, this integration poses challenges in terms of the discrete and continuous nature of the system. Concretely, models need to be developed that focus on the interaction of the ATO system with the safety system, and that consider modelling the influence of braking curves. Following this, suitable abstractions that involve reasoning over continuous data need to be explored and verification processes that scale for such a setting developed.

4 Formal Methods for AI

Formal methods have become a well-established and widely applied technique for ensuring the correctness of fundamental components of safety-critical systems in the railway domain, in particular verification techniques based on model checking (cf. Section 2.1). However, while a survey on software engineering for AI-based systems considered 248 studies published during the past decade, of which more than two-thirds since 2018 [83], the application of formal methods to AI-based systems is still in its infancy. A recent paper by Wing [105] lists the following three key insights:

1. The set of trustworthiness properties for AI systems, in contrast to traditional computing systems, needs to be extended beyond reliability, security, privacy, and usability to include properties such as probabilistic accuracy under uncertainty, fairness, robustness, accountability, and explainability.
2. To help ensure their trustworthiness, AI systems can benefit from the scrutiny of formal methods.
3. AI systems raise the bar on formal methods for two key reasons: the inherent probabilistic nature of machine-learned models, and the critical role of data in training, testing, and deploying a machine-learned model.

We envision to improve this situation by developing verification techniques that provide explainability or guarantees for AI-based systems in the specific safety-critical domain of railway systems, because, as Bešinović et al. put it, “although AI is still in its very infancy for the railway sector, there is certain evidence showing that its potential should not be underestimated” [24]. To this aim, we will first need to identify the state of the art of formal methods techniques developed for and applied to systems with AI-based components in the specific setting of transport systems (railways, but also automotive [97, 99]) and of safety certification. A key difference with respect to the traditional formal verification approach, i.e., verifying a correctness property specified in some logic over a system model, is the inherent probabilistic nature of the (machine-learned) model in case of AI-based systems. Based on our previous experiences, we intend to study how to deal with safety concerns in the presence of uncertainty, and how probabilistic (and statistical) model-checking techniques or correctness-by-construction techniques can be adapted to provide appropriate fail-safe guarantees for AI-based railway systems. Correctness-by-construction, in particular when considering also non-functional properties (X-by-construction), in combination with probability and runtime verification is being studied also at ISoLA [17, 18]. The same holds for formal methods for AI [76].

4.1 Guaranteeing Safety Behaviour

The wide availability of AI technologies and the pace of their evolution makes it hard for industry, with its consolidated processes, to profit from the potential benefits offered by these techniques. This is particularly true for the railway domain, in which the safety culture is strong, thus reinforcing the attachment to traditional, well-established practices that have proven their relevance even for the development of software systems that are not safety-critical. While many railway systems are required to fulfil SIL-4 certification requirements, many others, most notably maintenance systems, could instead make full use of the benefit of AI and process or data mining techniques. At the same time, explainability always must be ensured, to guarantee that the system behaviour can be explained, also for legal reasons, in case a failure occurs. Additionally, correctness is desirable as it would be difficult for a human to intervene and rectify mistakes made by such a system. For example, if a train is incorrectly routed across a junction it may take some time before it can be routed back across the junction in the correct direction and this would have impact on other trains in the area.

A concrete example from industry concerns an AI scheduler. Modern railway control systems are equipped with automatic route setting and traffic management but it is not clear how well they perform, specifically in case of a divergence from operational norms (e.g., if a train breaks down). It is typically the task of a qualified human to ensure trains run according to schedule, and to intervene when problems occur. Initial implementations of an AI scheduler could provide the human with guidance is such cases, assisting the human signaller in an efficient fashion to return the railway to an operational state by rerouting trains around the problem. When assisted by such an AI component, it is natural to want solutions that are presented by the AI component to both be explained and justified, and to not lead to intervention from safety critical components (for example, by providing solutions that violate the rules of the governing interlocking). For the human in the loop case it is essential for the AI system to be explainable and produce a justification that can be manually checked prior to making a decision.

4.2 Learning Formal Models of Railway Behaviour

Formal behavioural models are the building blocks of automated verification techniques in the field of formal methods. Such models define how a system behaves as a result of interacting with its users and its environment. Traditionally, these models (e.g., variants of automata and state machines) are obtained starting from semi-formal models (e.g., UML and message sequence charts) developed during the initial development phase. However, it frequently occurs that such behavioural models are either unavailable or outdated and thus need to be reconstructed from implementations in order to enable formal analysis. In such cases, model learning is an automated technique that can produce such models. This is a popular research field and much progress has been made since Vaandrager noted that "even though model learning has been applied successfully in several domains, the field is still in its infancy" [101]. Recent examples include [4, 39]. However, to the best of our knowledge, specific success stories in the railway domain are missing.

We envision the usage of data or process mining techniques to build digital twins of railway systems that can provide predictive (runtime) behaviour and ultimately enable real-time predictive monitoring and maintenance (cf. Section 4.3). Engineering digital twins is being studied also at ISoLA [55, 82, 56], including some initial, recent attempts in the railway domain [77]. This requires the use of a variety of techniques from formal methods, in particular probabilistic (and statistical) model checking to deal with the inherent probabilistic nature of the (machine-learned) model, game theory (for instance for controller synthesis), and automata or model learning, but also specific techniques from data or process mining [1]. Railway system models are characterised by the need to deal with real-time aspects and a degree of uncertainty. We will thus have to study how to perform data or process mining on the provided observation data, like execution traces (i.e., logs) of a railway system, and how to use this to learn a digital twin of the railway system. This digital twin is meant to be a formal model that can handle real-time and probabilistic or stochastic behaviour (e.g., conform to the timed stochastic models accepted by the Uppaal model checker).

4.3 AI for Monitoring and Maintenance

Current railway monitoring and maintenance systems are mostly rule-based and typically do not include AI-based components, which can be particularly useful, e.g., to predict possible failures and

to plan specific maintenance actions [33, 85]. AI systems, combined with existing rules provided by experts, can enable predictive maintenance, by identifying patterns of faults based on systems logs. Model learning for maintenance implies learning of the system model, and learning of a system's fault model, so that future faults can be predicted based on current system behaviour. The system's digital twin can also be used to forecast and simulate future long-term scenarios, thus helping to plan for maintenance actions in advance, i.e., before faults occur. Refactoring current rule-based maintenance systems with AI-based components poses numerous challenges. The effective exploitation of AI and process or data mining techniques requires the domain experts to annotate field data, such that the machine can learn from experts. Similarly, experts are needed to assess the correct behaviour and interpretation of possible failures of the AI-based maintenance system. Explainability of these systems becomes crucial, as well as correct communication of the behaviour's explanation to experts and other railway stakeholders. In the envisioned maintenance process, the behaviour of the onboard system can easily be reconstructed and visualised, and maintenance and improvement actions can be taken in a more flexible and effective way.

4.4 AI for Optimisation in Scheduling and Design

Whilst so far we addressed principal problems that need to be solved when using AI techniques, we now want to look at several specific applications in the railway domain. The first two applications concern problems for which currently no general optimal solutions exist, but where one can hope with the use of AI to achieve better solutions, i.e., solutions which are more efficient in terms of energy/time or which require less track-side equipment. For the solutions in this section it is, of course, essential that they are still safe and fulfil all safety requirements. Therefore it is anticipated that the solutions produced are not only very good, but that they also come with a guarantee or explanation. Overall, optimisations like, saving energy or requiring less track-side equipment contribute to the aim of reducing carbon. Also capacity improvements support the green deal (indirectly), as efficient and safe provision leads to a higher customer satisfaction and a higher uptake of railway use.

Our first application refers to scheduling which has frequently been considered as an optimisation problem in the past [71, 72, 42, 84, 89]. An AI Scheduler (cf. Section 4.1) could help to support and improve the decision making of a human signaller and optimising the flow of trains through the railway network.

A second standard application concerns the optimisation of railway design and layout. Railways are designed by engineers who create scheme plans with the topology of the railway and the layout of the equipment along the tracks. Published solutions for the automatic generation of signalling design seem not to consider optimisation at all. Desired would be an AI solution for the placement of the equipment that still fulfils all required constraints, such as number of balises in a given area, or a requested distance between balise groups, etc. Model checking/SMT solving can then be used to check the constraints and highlight counterexamples in a efficient way (where the visualisation of counterexamples in a domain specific area constitutes an interesting problem on its own.) Various AI techniques as well as Game theory and Explainable AI (XAI) can be used for the optimisation according to a given measure such as energy/material consumption.

5 Conclusion and Further Work

In this article, we discussed several challenges for Formal Methods in Railway linked to the areas of (1) Verification and Certification, (2) Modelling of ETCS related systems, and, (3) the use of AI in the railway domain. Their common aim is a robust development of complex railway technology that is reliable and efficient at the same time. Our specific focus in (1) and (3) was that any (new) techniques need to come with explanations/ guarantees in order to be accepted by the Railway engineers and the general public. Specifically, regarding the question of using AI techniques in Railway, further research is needed about which AI techniques we can apply and how to create explanations and guarantees. This is relatively straightforward in the case of applying, for instance, SMT-solving. Here, SMT-solving would provide counterexamples, and the gap to be closed concentrates on making these counterexamples (1) readable in the Railway context and (2) independently verifiable. Conversely, at the other end of the spectrum, if we want to apply machine-learning techniques for, e.g., classification, the situation is completely different: Explainable AI (XAI), with LIME [95] and SHAP [78] as prominent techniques,

has become established for providing explanations, however recent work has demonstrated that different XAI techniques do not necessarily coincide on their results [43]. Problems like these prompt research towards a theory of faithful explanations [93] and the idea of Verifiable XAI [49]. More effort and case studies will be needed to develop these techniques and make them usable in the Railway context.

Acknowledgment We would like to thank the anonymous referees for their constructive criticism and helpful comments.

References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4
2. ADLINK Technology: Transforming the rail industry with artificial intelligence (2021), https://www.globalrailwayreview.com/whitepaper/127609/transforming-the-rail-industry-with-ai
3. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans. Model. Comput. Simul. **28**(1), 6:1–6:39 (2018). https://doi.org/10.1145/3158668
4. Aichernig, B.K., Bloem, R., Ebrahimi, M., Horn, M., Pernkopf, F., Roth, W., Rupp, A., Tappler, M., Tranninger, M.: Learning a behavior model of hybrid systems through combining model-based testing and machine learning. In: Gaston, C., Kosmatov, N., Gall, P.L. (eds.) ICTSS 2019. LNCS, vol. 11812, pp. 3–21. Springer (2019). https://doi.org/10.1007/978-3-030-31280-0_1
5. Anselmi, A., Bernardeschi, C., Fantechi, A., Gnesi, S., Larosa, S., Mongardi, G., Torielli, F.: An experience in formal verification of safety properties of a railway signalling control system. In: Rabe, G. (ed.) Proceedings of the 14th International Conference on Computer Safety, Reliability and Security (SAFECOMP 1995). pp. 474–488. Springer (1995). https://doi.org/10.1007/978-1-4471-3054-3_33
6. Apt, K.R., Kozen, D.: Limits for automatic verification of finite-state concurrent systems. Inf. Process. Lett. **22**(6), 307–309 (1986)
7. Arcaini, P., Kofroň, J., Ježek, P.: Validation of the hybrid ERTMS/ETCS level 3 using SPIN. Int. J. Softw. Tools Technol. Transf. **22**(3), 265–279 (2020). https://doi.org/10.1007/s10009-019-00539-x
8. Ashok, P., Jackermeier, M., Kretínský, J., Weinhuber, C., Weininger, M., Yadav, M.: dtControl 2.0: Explainable strategy representation via decision tree learning steered by experts. In: TACAS 2021. LNCS, vol. 12652, pp. 326–345. Springer (2021). https://doi.org/10.1007/978-3-030-72013-1_17
9. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press (2008)
10. Bartholomeus, M., Luttik, B., Willemse, T.A.C.: Modelling and analysing ERTMS hybrid level 3 with the mCRL2 toolset. In: Howar, F., Barnat, J. (eds.) FMICS 2018. LNCS, vol. 11119, pp. 98–114. Springer (2018). https://doi.org/10.1007/978-3-030-00244-2_7
11. Basile, D., ter Beek, M.H., Ciancia, V.: Statistical model checking of a moving block railway signalling scenario with UPPAAL SMC. In: Margaria, T., Steffen, B. (eds.) ISoLA 2018. LNCS, vol. 11245, pp. 372–391. Springer (2018). https://doi.org/10.1007/978-3-030-03421-4_24
12. Basile, D., ter Beek, M.H., Ferrari, A., Legay, A.: Modelling and Analysing ERTMS L3 Moving Block Railway Signalling with Simulink and UPPAAL SMC. In: Larsen, K.G., Willemse, T. (eds.) FMICS 2019. LNCS, vol. 11687, pp. 1–21. Springer (2019). https://doi.org/10.1007/978-3-030-27008-7_1
13. Basile, D., ter Beek, M.H., Legay, A.: Strategy synthesis for autonomous driving in a moving block railway system with UPPAAL STRATEGO. In: Gotsman, A., Sokolova, A. (eds.) FORTE 2020. LNCS, vol. 12136, pp. 3–21. Springer (2020). https://doi.org/10.1007/978-3-030-50086-3_1
14. Basile, D., Fantechi, A., Rosadi, I.: Formal analysis of the UNISIG safety application intermediate sublayer: Applying formal methods to railway standard interfaces. In: Lluch-Lafuente, A., Mavridou, A. (eds.) FMICS 2021. LNCS, vol. 12863, pp. 174–190. Springer (2021). https://doi.org/10.1007/978-3-030-85248-1_11
15. Basile, D., Fantechi, A., Rucher, L., Mandò, G.: Analysing an autonomous tramway positioning system with the UPPAAL statistical model checker. Form. Asp. Comp. **33**(6), 957–987 (2021). https://doi.org/10.1007/s00165-021-00556-1
16. Basten, T., Bol, R.N., Voorhoeve, M.: Simulating and analyzing railway interlockings in ExSpect. IEEE Parallel Distributed Technol. Syst. Appl. **3**(3), 50–62 (1995). https://doi.org/10.1109/M-PDT.1995.414843
17. ter Beek, M.H., Cleophas, L., Legay, A., Schaefer, I., Watson, B.W.: X-by-Construction: Correctness meets probability. In: Margaria, T., Steffen, B. (eds.) ISoLA 2020. LNCS, vol. 12476, pp. 211–215. Springer (2020). https://doi.org/10.1007/978-3-030-61362-4_11
18. ter Beek, M.H., Cleophas, L., Leucker, M., Schaefer, I.: X-by-Construction meets runtime verification. In: Margaria, T., Steffen, B. (eds.) ISoLA 2022. LNCS, Springer (2022)

19. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K.G., Lime, D.: UPPAAL-Tiga: Time for playing games! In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 121–125. Springer (2007). https://doi.org/10.1007/978-3-540-73368-3_14
20. Berger, U., James, P., Lawrence, A., Roggenbach, M., Seisenberger, M.: Verification of the European Rail Traffic Management System in real-time Maude. Sci. Comput. Program. **154**, 61–88 (2018). https://doi.org/10.1016/j.scico.2017.10.011
21. Berger, U., Lawrence, A., Forsberg, F.N., Seisenberger, M.: Extracting verified decision procedures: DPLL and resolution. Log. Methods Comp. Sci. **11**(1), 1–18 (2015). https://doi.org/10.2168/LMCS-11(1:6)2015
22. Bernardeschi, C., Fantechi, A., Gnesi, S., Larosa, S., Mongardi, G., Romano, D.: A formal verification environment for railway signaling system design. Formal Methods Syst. Des. **12**(2), 139–161 (1998). https://doi.org/10.1023/A:1008645826258
23. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions. Texts in Theoretical Computer Science. An EATCS Series, Springer (2004). https://doi.org/10.1007/978-3-662-07964-5
24. Bešinović, N., Donato, L.D., Flammini, F., Goverde, R.M.P., Lin, Z., Liu, R., Marrone, S., Nardone, R., Tang, T., Vittorini, V.: Artificial intelligence in railway transport: Taxonomy, regulations and applications. IEEE Trans. Intell. Transp. Syst. (2022). https://doi.org/10.1109/TITS.2021.3131637
25. Biagi, M., Carnevali, L., Paolieri, M., Vicario, E.: Performability evaluation of the ERTMS/ETCS – level 3. Transp. Res. C-Emer. **82**, 314–336 (2017). https://doi.org/10.1016/j.trc.2017.07.002
26. Blanchette, J.C., Fleury, M., Lammich, P., Weidenbach, C.: A verified SAT solver framework with learn, forget, restart, and incrementality. J. Autom. Reason. **61**(1-4), 333–365 (2018). https://doi.org/10.1007/s10817-018-9455-7
27. Bouwman, M., Janssen, B., Luttik, B.: Formal modelling and verification of an interlocking using mCRL2. In: Larsen, K.G., Willemse, T.A.C. (eds.) FMICS 2019. LNCS, vol. 11687, pp. 22–39. Springer (2019). https://doi.org/10.1007/978-3-030-27008-7_2
28. Bouwman, M., van der Wal, D., Luttik, B., Stoelinga, M., Rensink, A.: A case in point: Verification and testing of a EULYNX interface. Form. Asp. Comput. (2022). https://doi.org/10.1145/3528207
29. Bouyer, P., Laroussinie, F., Reynier, P.: Diagonal constraints in timed automata: Forward analysis of timed systems. In: Pettersson, P., Yi, W. (eds.) FORMATS 2005. LNCS, vol. 3829, pp. 112–126. Springer (2005). https://doi.org/10.1007/11603009_10
30. Brunner, J., Lammich, P.: Formal verification of an executable LTL model checker with partial order reduction. J. Autom. Reason. **60**(1), 3–21 (2018). https://doi.org/10.1007/s10817-017-9418-4
31. Butler, M.J., Hoang, T.S., Raschke, A., Reichl, K.: Introduction to special section on the ABZ 2018 case study: Hybrid ERTMS/ETCS level 3. Int. J. Softw. Tools Technol. Transf. **22**(3), 249–255 (2020). https://doi.org/10.1007/s10009-020-00562-3
32. Cappart, Q., Limbrée, C., Schaus, P., Quilbeuf, J., Traonouez, L., Legay, A.: Verification of interlocking systems using statistical model checking. In: Proceedings of the 18th International Symposium on High Assurance Systems Engineering (HASE 2017). pp. 61–68. IEEE (2017). https://doi.org/10.1109/HASE.2017.10
33. Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., da Piedade Francisco, R., Basto, J.P.T.V., Alcalá, S.G.S.: A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. **137** (2019). https://doi.org/10.1016/j.cie.2019.106024
34. Chiappini, A., Cimatti, A., Macchi, L., Rebollo, O., Roveri, M., Susi, A., Tonetta, S., Vittorini, B.: Formalization and validation of a subset of the European Train Control System. In: Proceedings of the 32nd International Conference on Software Engineering (ICSE 2010). pp. 109–118. ACM (2010). https://doi.org/10.1145/1810295.1810312
35. Cimatti, A., Corvino, R., Lazzaro, A., Narasamdya, I., Rizzo, T., Roveri, M., Sanseviero, A., Tchaltsev, A.: Formal verification and validation of ERTMS industrial railway train spacing system. In: Madhusudan, P., Seshia, S.A. (eds.) CAV 2012. LNCS, vol. 7358, pp. 378–393. Springer (2012). https://doi.org/10.1007/978-3-642-31424-7_29
36. Cimatti, A., Giunchiglia, F., Mongardi, G., Romano, D., Torielli, F., Traverso, P.: Model checking safety critical software with SPIN: an application to a railway interlocking system. In: Ehrenberger, W.D. (ed.) SAFECOMP 1998. LNCS, vol. 1516, pp. 284–295. Springer (1998). https://doi.org/10.1007/3-540-49646-7_22
37. Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.): Handbook of Model Checking. Springer (2018). https://doi.org/10.1007/978-3-319-10575-8
38. Cunha, A., Macedo, N.: Validating the hybrid ERTMS/ETCS level 3 concept with Electrum. Int. J. Softw. Tools Technol. Transf. **22**(3), 281–296 (2020). https://doi.org/10.1007/s10009-019-00540-4
39. Damasceno, C.D.N., Mousavi, M.R., da Silva Simão, A.: Learning by sampling: learning behavioral family models from software product lines. Empir. Softw. Eng. **26**(1), 4:1–4:46 (2021). https://doi.org/10.1007/s10664-020-09912-w

40. David, A., Jensen, P.G., Larsen, K.G., Mikucionis, M., Taankvist, J.H.: Uppaal Stratego. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 206–211. Springer (2015). https://doi.org/10.1007/978-3-662-46681-0_16
41. Dghaym, D., Dalvandi, M., Poppleton, M., Snook, C.F.: Formalising the hybrid ERTMS level 3 specification in iUML-B and Event-B. Int. J. Softw. Tools Technol. Transf. **22**(3), 297–313 (2020). https://doi.org/10.1007/s10009-019-00548-w
42. Dillmann, S., Hähnle, R.: Automated planning of ETCS tracks. In: Dutilleul, S.C., Lecomte, T., Romanovsky, A.B. (eds.) RSSRAIL 2019. LNCS, vol. 11495, pp. 79–90. Springer (2019). https://doi.org/10.1007/978-3-030-18744-6_5
43. Duell, J., Fan, X., Burnett, B., Aarts, G., Zhou, S.M.: A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records. In: Proceedings of the 7th EMBS International Conference on Biomedical and Health Informatics (BHI 2021). pp. 1–4. IEEE (2021). https://doi.org/10.1109/BHI50953.2021.9508618
44. Eisner, C.: Using symbolic CTL model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. Int. J. Softw. Tools Technol. Transf. **4**(1), 107–124 (2002). https://doi.org/10.1007/s100090100063
45. Esparza, J., Lammich, P., Neumann, R., Nipkow, T., Schimpf, A., Smaus, J.: A fully verified executable LTL model checker. Arch. Formal Proofs (May 2014), https://isa-afp.org/entries/CAVA_LTL_Modelchecker.html
46. European Committee for Electrotechnical Standardization: CENELEC EN 50128 — Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems (June 2011)
47. European Committee for Electrotechnical Standardization: CENELEC EN 50126-1 — Railway applications – The specification and demonstration of reliability, availability, maintainability and safety (RAMS) – Part 1: Generic RAMS process (October 2017)
48. European Committee for Electrotechnical Standardization: CENELEC EN 50129 — Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling (November 2018)
49. Fan, X.: Verifiable Explainable AI (2021), unpublished manuscript
50. Fantechi, A.: Distributing the challenge of model checking interlocking control tables. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012. LNCS, vol. 7610, pp. 276–289. Springer (2012). https://doi.org/10.1007/978-3-642-34032-1_26
51. Ferrari, A., ter Beek, M.H.: Formal methods in railways: a systematic mapping study. ACM Comput. Surv. (2022). https://doi.org/10.1145/3520480
52. Ferrari, A., Magnani, G., Grasso, D., Fantechi, A.: Model checking interlocking control tables. In: Schnieder, E., Tarnai, G. (eds.) Proceedings of the 8th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 2010), pp. 107–115. Springer (2010). https://doi.org/10.1007/978-3-642-14261-1_11
53. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H.: Systematic evaluation and usability analysis of formal tools for railway system design. IEEE Trans. Softw. Eng. (2021). https://doi.org/10.1109/TSE.2021.3124677
54. Ferrari, A., Mazzanti, F., Basile, D., ter Beek, M.H., Fantechi, A.: Comparing formal tools for system design: a judgment study. In: Proceedings of the 42nd International Conference on Software Engineering (ICSE 2020). pp. 62–74. ACM (2020). https://doi.org/10.1145/3377811.3380373
55. Fitzgerald, J.S., Larsen, P.G., Margaria, T., Woodcock, J.: Engineering of digital twins for cyber-physical systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2020. LNCS, vol. 12479, pp. 49–53. Springer (2020). https://doi.org/10.1007/978-3-030-83723-5_4
56. Fitzgerald, J.S., Larsen, P.G., Margaria, T., Woodcock, J., Gomes, C.: Digital twin engineering. In: Margaria, T., Steffen, B. (eds.) ISoLA 2022. LNCS, Springer (2022)
57. Fotso, S.J.T., Frappier, M., Laleau, R., Mammar, A.: Modeling the hybrid ERTMS/ETCS level 3 standard using a formal requirements engineering approach. Int. J. Softw. Tools Technol. Transf. **22**(3), 349–363 (2020). https://doi.org/10.1007/s10009-019-00542-2
58. Furness, N., van Houten, H., Arenas, L., Bartholomeus, M.: ERTMS level 3: the game-changer. IRSE News **232**, 2–9 (April 2017), https://www.irse.nl/resources/170314-ERTMS-L3-The-gamechanger-from-IRSE-News-Issue-232.pdf
59. Gossen, F., Margaria, T., Steffen, B.: Towards explainability in machine learning: The formal methods way. IT Prof. **22**(4), 8–12 (2020). https://doi.org/10.1109/MITP.2020.3005640
60. Griggio, A., Roveri, M., Tonetta, S.: Certifying proofs for SAT-based model checking. Formal Methods Syst. Des. **57**(2), 178–210 (2021). https://doi.org/10.1007/s10703-021-00369-1
61. Groote, J.F., Vlijmen, S.F.M., Koorn, J.W.C.: The safety guaranteeing system at station Hoorn-Kersenboogerd. In: Proceedings of the 10th Annual Conference on Computer Assurance Systems Integrity, Software Safety and Process Security (COMPASS 1995). pp. 57–68. IEEE (1995). https://doi.org/10.1109/CMPASS.1995.521887

62. Hansen, D., Leuschel, M., Körner, P., Krings, S., Naulin, T., Nayeri, N., Schneider, D., Skowron, F.: Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model. Int. J. Softw. Tools Technol. Transf. **22**(3), 315–332 (2020). https://doi.org/10.1007/s10009-020-00551-6
63. Hartonas-Garmhausen, V., Kurfess, T.R., Clarke, E.M., Long, D.E.: Automatic verification of industrial designs. In: Proceedings of the Workshop on Industrial-Strength Formal Specification Techniques (WIFT 1995). pp. 88–96. IEEE Computer Society (1995). https://doi.org/10.1109/WIFT.1995.515481
64. Haxthausen, A.E., Kjær, A.A., Bliguet, M.L.: Formal development of a tool for automated modelling and verification of relay interlocking systems. In: Butler, M.J., Schulte, W. (eds.) FM 2011. LNCS, vol. 6664, pp. 118–132. Springer (2011). https://doi.org/10.1007/978-3-642-21437-0_11
65. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. Syst. Sci. **57**(1), 94–124 (1998). https://doi.org/10.1006/jcss.1998.1581
66. Hong, L.V., Haxthausen, A.E., Peleska, J.: Formal modelling and verification of interlocking systems featuring sequential release. Sci. Comput. Program. **133**, 91–115 (2017). https://doi.org/10.1016/j.scico.2016.05.010
67. Hvilshøj, F., Iosifidis, A., Assent, I.: ECINN: Efficient counterfactuals from invertible neural networks. CoRR **abs/2103.13701** (2021), https://doi.org/10.48550/arXiv.2103.13701
68. James, P., Lawrence, A., Moller, F., Roggenbach, M., Seisenberger, M., Setzer, A., Kanso, K., Chadwick, S.: Verification of solid state interlocking programs. In: Counsell, S., Núñez, M. (eds.) SEFM 2013. LNCS, vol. 8368, pp. 253–268. Springer (2014). https://doi.org/10.1007/978-3-319-05032-4_19
69. James, P., Moller, F., Nga, N.H., Roggenbach, M., Schneider, S.A., Treharne, H.: Techniques for modelling and verifying railway interlockings. Int. J. Softw. Tools Technol. Transf. **16**(6), 685–711 (2014). https://doi.org/10.1007/s10009-014-0304-7
70. James, P., Moller, F., Nguyen, H.N., Roggenbach, M., Schneider, S., Treharne, H.: On modelling and verifying railway interlockings: Tracking train lengths. Sci. Comput. Program. **96**, 315–336 (2014). https://doi.org/10.1016/j.scico.2014.04.005
71. James, P., Moller, F., Nguyen, H.N., Roggenbach, M., Schneider, S.A., Treharne, H., Trumble, M., Williams, D.M.: Verification of scheme plans using CSP || B. In: Counsell, S., Núñez, M. (eds.) SEFM 2013 Workshops. LNCS, vol. 8368, pp. 189–204. Springer (2014). https://doi.org/10.1007/978-3-319-05032-4_15
72. James, P., Roggenbach, M.: Encapsulating formal methods within domain specific languages: A solution for verifying railway scheme plans. Math. Comput. Sci. **8**(1), 11–38 (2014). https://doi.org/10.1007/s11786-014-0174-0
73. Kant, G., Laarman, A., Meijer, J., van de Pol, J., Blom, S., van Dijk, T.: LTSmin: High-performance language-independent model checking. In: Baier, C., Tinelli, C. (eds.) FORMATS 2015. LNCS, vol. 9035, pp. 692–707. Springer (2015). https://doi.org/10.1007/978-3-662-46681-0_61
74. Katz, G., Barrett, C.W., Tinelli, C., Reynolds, A., Hadarean, L.: Lazy proofs for DPLL(T)-based SMT solvers. In: Proceedings of the 16th Conference on Formal Methods in Computer-Aided Design (FMCAD 2016). pp. 93–100. IEEE (2016). https://doi.org/10.1109/FMCAD.2016.7886666
75. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: PRISM-games 3.0: Stochastic game verification with concurrency, equilibria and time. In: Lahiri, S.K., Wang, C. (eds.) CAV 2020. LNCS, vol. 12225, pp. 475–487. Springer (2020). https://doi.org/10.1007/978-3-030-53291-8_25
76. Larsen, K.G., Legay, A., Steffen, B., Stoelinga, M.: Formal methods meet machine learning. In: Margaria, T., Steffen, B. (eds.) ISoLA 2022. LNCS, Springer (2022)
77. Lecomte, T.: Digital modelling in the railways. In: Margaria, T., Steffen, B. (eds.) ISoLA 2020. LNCS, vol. 12479, pp. 124–139. Springer (2020). https://doi.org/10.1007/978-3-030-83723-5_9
78. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017). pp. 4768–4777 (2017), https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
79. Macedo, H.D., Fantechi, A., Haxthausen, A.E.: Compositional verification of multi-station interlocking systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2016. LNCS, vol. 9953, pp. 279–293 (2016). https://doi.org/10.1007/978-3-319-47169-3_20
80. Macedo, H.D., Fantechi, A., Haxthausen, A.E.: Compositional model checking of interlocking systems for lines with multiple stations. In: Barrett, C.W., Davies, M., Kahsai, T. (eds.) NFM 2017. LNCS, vol. 10227, pp. 146–162 (2017). https://doi.org/10.1007/978-3-319-57288-8_11
81. Mammar, A., Frappier, M., Tueno Fotso, S.J., Laleau, R.: A formal refinement-based analysis of the hybrid ERTMS/ETCS level 3 standard. Int. J. Softw. Tools Technol. Transf. **22**(3), 333–347 (2020). https://doi.org/10.1007/s10009-019-00543-1
82. Margaria, T., Schieweck, A.: Towards engineering digital twins by active behaviour mining. In: Olderog, E., Steffen, B., Yi, W. (eds.) Model Checking, Synthesis, and Learning. LNCS, vol. 13030, pp. 138–163. Springer (2021). https://doi.org/10.1007/978-3-030-91384-7_8
83. Martínez-Fernández, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., Vollmer, A.M., Wagner, S.: Software engineering for AI-based systems: A survey. ACM Trans. Softw. Eng. Methodol. **31**(2), 37e:1–37e:59 (2022). https://doi.org/10.1145/3487043

84. Mazzanti, F., Spagnolo, G.O., Longa, S.D., Ferrari, A.: Deadlock avoidance in train scheduling: A model checking approach. In: Lang, F., Flammini, F. (eds.) FMICS 2014. LNCS, vol. 8718, pp. 109–123. Springer (2014). https://doi.org/10.1007/978-3-319-10702-8_8
85. Nakhaee, M.C., Hiemstra, D., Stoelinga, M., van Noort, M.: The recent applications of machine learning in rail track maintenance: A survey. In: Dutilleul, S.C., Lecomte, T., Romanovsky, A.B. (eds.) RSSRAIL 2019. LNCS, vol. 11495, pp. 91–105. Springer (2019). https://doi.org/10.1007/978-3-030-18744-6_6
86. Namjoshi, K.S.: Certifying model checkers. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 2–13. Springer (2001). https://doi.org/10.1007/3-540-44585-4_2
87. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002). https://doi.org/10.1007/3-540-45949-9
88. Oortwijn, W., Huisman, M., Joosten, S.J.C., van de Pol, J.: Automated verification of parallel nested DFS. In: Biere, A., Parker, D. (eds.) FORMATS 2020. LNCS, vol. 12078, pp. 247–265. Springer (2020). https://doi.org/10.1007/978-3-030-45190-5_14
89. Peham, T., Przigoda, J., Przigoda, N., Wille, R.: Optimal railway routing using virtual subsections. In: Dutilleul, S.C., Haxthausen, A.E., Lecomte, T. (eds.) RSSRAIL 2022. LNCS, vol. 13294, pp. 63–79. Springer (2022). https://doi.org/10.1007/978-3-031-05814-1_5
90. Peleska, J., Haxthausen, A.E., Lecomte, T.: Standardisation considerations for autonomous train control. In: Margaria, T., Steffen, B. (eds.) ISoLA 2022. LNCS, Springer (2022), in this volume
91. Platzer, A., Quesel, J.: European train control system: A case study in formal verification. In: Breitman, K.K., Cavalcanti, A. (eds.) ICFEM 2009. LNCS, vol. 5885, pp. 246–265. Springer (2009). https://doi.org/10.1007/978-3-642-10373-5_13
92. van de Pol, J.C.: Automated verification of nested DFS. In: Núñez, M., Güdemann, M. (eds.) FMICS 2015. LNCS, vol. 9128, pp. 181–197. Springer (2015). https://doi.org/10.1007/978-3-319-19458-5_12
93. Potyka, N., Yin, X., Toni, F.: Towards a theory of faithfulness: Faithful explanations of differentiable classifiers over continuous data. CoRR **abs/2205.09620** (2022), https://doi.org/10.48550/arXiv.2205.09620
94. Pranger, S., Könighofer, B., Posch, L., Bloem, R.: TEMPEST – Synthesis tool for reactive systems and shields in probabilistic environments. In: Hou, Z., Ganesh, V. (eds.) ATVA 2021. LNCS, vol. 12971, pp. 222–228. Springer (2021). https://doi.org/10.1007/978-3-030-88885-5_15
95. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the Demonstrations Session of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016). pp. 97–101 (2016). https://doi.org/10.18653/v1/n16-3020
96. Ringer, T., Palmskog, K., Sergey, I., Gligoric, M., Tatlock, Z.: QED at large: A survey of engineering of formally verified software. Found. Trends Program. Lang. **5**(2-3), 102–281 (2019). https://doi.org/10.1561/2500000045
97. Shafaei, S., Kugele, S., Osman, M.H., Knoll, A.C.: Uncertainty in machine learning: A safety perspective on autonomous driving. In: Gallina, B., Skavhaug, A., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2018. LNCS, vol. 11094, pp. 458–464. Springer (2018). https://doi.org/10.1007/978-3-319-99229-7_39
98. Siegel, S.F.: What's wrong with on-the-fly partial order reduction. In: Dillig, I., Tasiran, S. (eds.) CAV 2019. LNCS, vol. 11562, pp. 478–495. Springer (2019). https://doi.org/10.1007/978-3-030-25543-5_27
99. Tuncali, C.E., Fainekos, G., Prokhorov, D.V., Ito, H., Kapinski, J.: Requirements-driven test generation for autonomous vehicles with machine learning components. IEEE Trans. Intell. Veh. **5**(2), 265–280 (2020). https://doi.org/10.1109/TIV.2019.2955903
100. Underwriters Laboratories Inc.: ANSI/UL 4600 Standard for Safety Evaluation of Autonomous Products (March 2022)
101. Vaandrager, F.W.: Model learning. Commun. ACM **60**(2), 86–95 (2017). https://doi.org/10.1145/2967606
102. Wiedijk, F. (ed.): The Seventeen Provers of the World. LNCS 3600, Springer (2006). https://doi.org/10.1007/11542384
103. Wimmer, S., Herbreteau, F., van de Pol, J.: Certifying emptiness of timed Büchi automata. In: Bertrand, N., Jansen, N. (eds.) FORMATS 2020. LNCS, vol. 12288, pp. 58–75. Springer (2020). https://doi.org/10.1007/978-3-030-57628-8_4
104. Wimmer, S., Lammich, P.: Verified model checking of timed automata. In: Beyer, D., Huisman, M. (eds.) FORMATS 2018. LNCS, vol. 10805, pp. 61–78. Springer (2018). https://doi.org/10.1007/978-3-319-89960-2_4
105. Wing, J.M.: Trustworthy AI. Commun. ACM **64**(10), 64–71 (2021). https://doi.org/10.1145/3448248
106. Winter, K.: Optimising ordering strategies for symbolic model checking of railway interlockings. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012. LNCS, vol. 7610, pp. 246–260. Springer (2012). https://doi.org/10.1007/978-3-642-34032-1_24