# DESIGN AND AUTOMATIC GENERATION OF SAFETY CASES OF ML-ENABLED AUTONOMOUS DRIVING SYSTEMS

**MITHILA SIVAKUMAR**

**A THESIS SUBMITTED TO**
**THE FACULTY OF GRADUATE STUDIES**
**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**
**FOR THE DEGREE OF**
**MASTER OF SCIENCE**

**GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**
**YORK UNIVERSITY**
**TORONTO, ONTARIO**

**April 2024**

**Abstract**

Safety cases play a pivotal role in ensuring system reliability and acceptability, providing a structured argument supported by evidence. However, gaps in safety case literature hinder comprehensive safety assurance practices. In this thesis, we address this challenge through a three-fold approach. First, we conducted a bibliometric analysis following PRISMA 2020 guidelines to identify trends and knowledge gaps in safety assurance research. The analysis reveals critical areas lacking full safety cases and highlights the need for automated safety case construction. Then, we manually constructed a safety case for an ML-enabled component of an autonomous vehicle. Finally, leveraging large language models like GPT-4, we conducted experiments to automate safety case generation. Results indicate that GPT-4 produces safety cases with moderate accuracy and high semantic similarity to ground truth cases. This comprehensive methodology enhances safety practices, aiding researchers, analysts, and regulators in achieving robust safety assurance in complex systems.

**Acknowledgement**

I would like to sincerely thank my supervisor, Dr. Alvine Boaye Belle for her continuous support and guidance that has helped me complete my research project on time.

I would like to warmly thank my co-supervisor Dr. Jinjun Shan for providing access to his lab and for his invaluable suggestions. Dr.Shan has always motivated me to bring out the best in me and I appreciate his assistance throughout the course of my research project.

I would also like to thank Dr. Mingfeng Yuan, Dr. Shan's student, for providing access to his research that has enabled me to perform my research efficiently.

I would like to thank my defense committee chair, Dr. Marios Fokaefs for always providing invaluable comments that helped improved the quality of my thesis.

I would like to extend my warmest regards to Dr. Mannar Jammal for agreeing to be the external member of my thesis committee.

Finally, I would like to thank my colleagues, EECS department and Lassonde school of engineering for giving me the invaluable opportunity to complete my thesis.

# Contents

# List of Tables

# List of Figures

**Abbreviations**

1. ML - Machine Learning
2. AI - Artificial Intelligence
3. RL - Reinforcement Learning
4. CPS - Cyber-Physical Systems
5. GSN - Goal Structuring Notation
6. SAC - Safety Assurance Case
7. LLM - Large Language Model
8. PRISMA - Preferred Reporting Items for Systematic Reviews and Meta-Analyses
9. SEGRESS - Software Engineering Guidelines for Reporting Secondary Studies
10. AMLAS - Assurance of Machine Learning for use in Autonomous Systems
11. HARA - Hazard Analysis and Risk Assessment
12. HAZOP - Hazard and Operability
13. ASIL - Automotive Safety Integrity Level
14. TNR - Tire Noise Recognition
15. LMS - Lane Management System

# 1 Introduction, Motivation and Contributions

## 1.1 Context

Safety incidents involving autonomous vehicles and other safety-critical systems can lead to fatal outcomes and pose significant risks to manufacturers and the public [9]. Recent tragedies, like the implosion of the Titan submersible [10], highlight the need for prioritizing safety and complying with industry standards. In order to prevent such disasters, safety-critical systems (e.g., cyber-physical systems) are subject to regulation through domain-specific safety standards such as ISO 26262, IEC 61508, and DO-178C [11, 12, 13, 14, 15]. To certify these systems, regulatory authorities have established different means by which the most common practice documents safety cases. Enforcing mandatory safety measures and utilizing safety cases can prevent such accidents and safeguard lives. Safety assurance is essential across several industries such as nuclear energy, aerospace, and transportation. It usually involves building a convincing argument that aims at demonstrating systems are safe enough to operate in specific environments.

Thus, the use of safety cases allows verifying the correct implementation of the desired systems' properties (or requirements). This allows preventing the failure of such systems. That failure may result in loss of life, severe injuries, large-scale environmental damage, property destruction, and major economic and capital loss [16]. Safety cases can be presented in various ways, including plain text such as structured prose or Trust cases [17], as well as through graphical notations [18]. These graphical representations encompass GSN and Claim-Argument-Evidence (CAE). Among these notations, GSN is widely favored for depicting safety cases [18].

Our thesis is threefold. First, we conduct a secondary study, more specifically, a bibliometric analysis on the subject of safety cases with the purpose of understanding its overall intellectual landscape. We chose to focus our analysis on the last ten years since, by concentrating on the most recent decade, we can analyze and categorize the most current developments and trends in safety cases that are most relevant to contemporary challenges and practices. To achieve this, we utilize descriptive statistics to analyze primary studies, authors, venues, and other relevant information gathered from the literature. The synthesis provides valuable insights into the characteristics and dynamics of safety case research, contributing to a comprehensive understanding of the field's current state and outlining potential future research directions. For this study, we selected 224 primary studies relevant to safety cases, from the period spanning 2012 to 2022. We relied on five well-established databases to select these studies.

A powerful approach for applying a set of quantitative metrics to evaluate the influence of academic contributions within a specific field is bibliometrics [19]. Bibliometrics is widely recognized as an effective method for uncovering the cumulative knowledge structures and intellectual associations within a specific field or discipline [19, 20]. Several bibliometric analyses (e.g., [19, 21, 22, 23]) have therefore been performed in various domains (e.g., energy, information technology, finance) and have received significant attention. Still, even though there is a rich body of literature on safety cases, there is — to the best of our knowledge — no secondary study that has performed a detailed bibliometric analysis to investigate the trends, patterns, and relationships characterizing that literature. This makes it challenging to get a holistic view of the safety case landscape and hinders the identification of the most promising future research directions in that area.

The bibliometric analysis we report in this paper will be of great assistance to researchers and practitioners (e.g., corporate safety analysts, and regulators) involved in the development and certification of mission-critical systems. More specifically, by providing valuable insights, trends, and patterns in the literature on safety cases, this analysis will guide them in exploring, implementing, and advancing safety assurance solutions. Our findings will help the audience understand the current state of safety case research and identify potential areas for improvement and innovation in safety assurance practices.

Secondly, we manually create a safety case following the frameworks of AMLAS [3] and SMIRK [2]. Access to complete safety cases is often limited in safety case literature for various reasons. Firstly, due to the critical nature of safety-critical systems, granting open access to the safety case may risk exposing sensitive system architecture to competitors or malicious actors. Secondly, manual creation of safety cases requires substantial time, effort, and a large team; for instance, the entire SMIRK project [3] spanned approximately 2 years. These challenges hinder researchers from accessing complete safety cases, impeding further experimentation and the development of new approaches in the field. To address this, our thesis aims to provide researchers and practitioners with access to a complete safety case, serving as a foundational resource for future endeavors in safety assurance.

Finally, we conduct experiments on GPT-4 to determine its ability to generate safety cases automatically. For this purpose, we utilize three safety cases from literature as ground truths. However, employing generative AI models like GPT-4 for software modeling, particularly safety case modeling, presents certain challenges. Firstly, concerns arise regarding the potential generation of hallucinated responses by GPT-4 [24, 21]. This implies that the safety arguments produced by the model might contain inaccuracies or even false information. In situations involving high-risk critical systems, such as autonomous vehicles, such inaccuracies could lead to severely adverse consequences. Secondly, GPT-4 operates in a non-deterministic manner. In practical terms, this means that it has the capability to provide different responses to the same queries during different runs. This non-determinism introduces an additional layer of complexity and unpredictability into the process of generating safety arguments using GPT-4.

## 1.2 Problem Statement

The motivation behind this research stems from the critical importance of safety assurance in various domains, particularly in safety-critical systems where the reliability and integrity of operations are paramount. However, accessing complete safety cases for study and analysis is often challenging due to factors such as proprietary concerns, time-intensive manual creation processes, and security considerations. This limitation obstructs the progress of research and innovation in safety assurance methodologies. Therefore, this study aims to address this gap by providing access to a complete safety case, facilitating further research, experimentation, and the development of novel approaches in safety assurance.

Safety cases often take the form of extensive documents, sometimes spanning several hundred pages [25]. For example, the safety case for an air traffic control system can comprise more than 500 pages and reference approximately 400 documents, primarily in natural language [25]. However, these safety cases are typically manually generated, which can be a laborious, error-prone, and time-consuming process, especially when dealing with complex and diverse systems like cyber-physical systems (CPS) [26, 27, 28]. Consequently,

there is a pressing need to explore innovative approaches for automating the generation of safety cases.
In the realm of safety case research, despite the extensive body of work dedicated to automating the generation of safety cases, our investigation, to the best of our knowledge, stands as a foundation in employing GPT-4 for this purpose. Consequently, the primary focus of our paper revolves around the automation of safety case creation through the utilization of GPT-4. The main objective of this endeavor is to alleviate the workload of safety case modelers, offering a novel approach to safety case development.
Drawing inspiration from [29], adapted for safety cases, we conducted two sets of experiments. The first set aimed to evaluate GPT-4's competence in addressing fundamental queries related to the Goal Structuring Notation (GSN), a notation used for representing safety cases. In the second set, we explored GPT-4's potential in generating safety cases. To achieve this, we utilized three existing safety cases as benchmarks and conducted four distinct experiments, each involving variations in domain knowledge and GSN syntax comprehension. These experiments were designed to assess GPT-4's effectiveness in generating safety arguments.

### 1.3 Research Objectives

To identify current industry gaps, we conducted a thorough literature review through the help of bibliometric analysis. This review highlighted the need for greater automation in creating safety cases, especially considering their extensive nature. Additionally, we recognized the significance of focusing on ML-enabled autonomous driving systems, which appear to be the future direction. Furthermore, this analysis revealed that there are very few complete safety cases accessible to researchers, in existing literature. Therefore, the core objective of this thesis is to create a safety case for an ML-enabled self-driving vehicle, particularly in the context of the unsignalized 4-way intersection scenario introduced in [1]. Furthermore, we also investigate the feasibility and effectiveness of leveraging Large Language Models (LLMs) like GPT-4 [30] to automate the generation of (partial) safety cases, compliant with goal structuring notation (GSN). This objective is pursued by addressing the following set of research questions (RQs):
**RQ1: What are the existing techniques and methodologies supporting safety case development, and what are the gaps and limitations in the current state-of-the-art practices?** To answer this question, we will undertake a thorough examination of the prevailing literature pertaining to techniques in safety case development. This comprehensive review will fulfill two crucial objectives: Firstly, it will facilitate our understanding of the diverse methodologies and approaches utilized in the realm of safety case development. Secondly, this critical analysis will aid in identifying areas where improvements and innovations are needed in the domain of safety cases.
**RQ2: Can a safety case be created manually following AMLAS for the system defined in Yuan et al. [1]?** To address this research question, we manually created a safety case following the guidelines presented in AMLAS for the system described in Yuan et al. [1]. One main limitation when it comes to safety cases is that there is only a handful of existing literature that provides access to a complete safety case. Without many examples, it is difficult to create a safety case or perform experiments with the same. That scarcity of complete safety cases may for instance hamper the development of ML-powered approaches that aim at using available safety cases to automatically learn the safety assurance process for purposes such as (semi-

)automating the generation of safety cases. The answer to this research question will notably stand as a foundation for researchers who want to create safety case for an already deployed ML-enabled self-driving vehicles but does not have a starting point.

**RQ3: Can GPT-4 successfully generate (partial) safety cases and what are the practical implications, limitations, and challenges of employing LLMs in this context?**

To validate the feasibility of our approach, we conducted experiments to gauge the ability of GPT-4 to automatically generate (partial) safety cases. These scenarios will serve as test cases, allowing us to assess the practicality, accuracy, and limitations of GPT-4 generated safety cases. Through our preliminary experiments, we aim to answer the following sub (RQs):

**RQ3.1: Is GPT-4 sufficiently proficient in the Goal Structuring Notation?** Similar to Chen et al.[31], we employed a set of 19 questions to answer our research inquiry. These questions are designed to evaluate GPT-4's comprehension of the structural and semantic rules that define the Goal Structuring Notation (GSN) and its ability to generate concise goal structures.

**RQ3.2: Is GPT-4 able to generate safety cases that are structurally, semantically, and reasonably correct?**

To address this research question, we utilized three safety cases from the literature as our ground-truths. These safety cases pertain to different application domains: one in the field of energy and the other two from the automotive sector.

## 1.4 Contributions

The contributions of our work are as follows:

- **Contribution 1**: We have conducted a bibliometric analysis on safety cases spanning 10 years (2012-2022) and report the gaps in safety case literature.
- **Contribution 2**: We have created a complete safety case for the system presented in Yuan et al. [1] and provide access to this completed safety case.
- **Contribution 3**: We have evaluated GPT-4's performance in generating safety cases, focusing on its structural correctness, semantic accuracy, and overall reasonability. For this purpose, we have used GPT-4 to generate 48 safety cases similar to the three ground-truth safety cases at hand.

In this thesis, the three research questions are intricately woven together to provide a comprehensive exploration of safety case development and evaluation in the context of ML-enabled autonomous systems. Initially, in our bibliometric analysis, we delve into the existing techniques and methodologies supporting safety case development, identifying gaps and limitations within the current state-of-the-art practices. From this analysis, two prominent gaps emerge: the scarcity of complete safety cases and the limited research in automatic safety case creation. To address these gaps, first, we undertake the manual creation of a safety case following AMLAS, bridging the gap between theory and practice. This hands-on approach not only fills the void of complete safety cases but also lays a solid foundation for subsequent inquiry. Building upon this

foundation, we then turn our attention to the cutting-edge realm of artificial intelligence, specifically exploring the potential of GPT-4 in automating safety case creation. Through this iterative process, each research question builds upon the preceding one, forming a cohesive narrative that not only addresses identified gaps but also bridges theoretical insights with practical applications to pave the way for safer and more robust autonomous systems.

## 1.5 Thesis Structure

The rest of the thesis is organized as follows. Section **??** lists all the abbreviations and definitions. Section 2 explains the background concepts. Section 3 talks about the related work. Section 4 explains the methodology. Section 5, 6, and 7 explains the results of the research questions. Section 8 lists the threat to validity. Section 9 provides discussion and lists the future work. Section 10 concludes the work. Appendix A lists all the 224 primary studies included in the bibliometric analysis. Appendix B provides the list of all conference/journal venues with the abbreviations.

# 2 Background

## 2.1 What is a safety case?

An assurance case is a "*set of audible claims, arguments, and evidence created to support the claim that a defined system/service will satisfy particular requirements*" [32, 33, 34]. An assurance case is a document that eases the exchange of information between various system stakeholders (e.g., suppliers, acquirers), and between the operator and regulator, where the knowledge regarding a system's requirements (e.g., safety, security, reliability) is convincingly conveyed [33]. Assurance cases are structured as a hierarchy of claims, with lower-level claims drawing on concrete evidence, and serving as evidence to justify claims higher in the hierarchy [35, 36]. Several standards (e.g., ISO 26262) advocate the use of assurance cases to support the certification of various systems, including CPSs such as autonomous systems [35, 37]. Depending on the target requirement, there are several types of assurance cases, e.g., security cases, safety cases, dependability cases [38, 39, 33, 40]. Safety cases are assurance cases that support the safety assurance process [41]. In this study, we focus on safety cases as safety is a life-critical requirement.

A safety case aims to construct a convincing and valid argument, demonstrating that a system is adequately safe for operation within a specific environment. safety cases have been employed for several years and are witnessing a growing trend in constructing safety arguments for safety-critical systems across various industries. Domains such as automotive, aerospace, nuclear energy, and healthcare have increasingly adopted safety cases as a means to demonstrate and justify the safety of their systems [24]. This surge in adoption highlights the importance of safety cases in assuring the safety of mission-critical systems, ensuring compliance with industry standards (e.g., ISO 26262, DO-178C, UL 4600), and mitigating potential safety risks. Safety-critical systems (e.g., CPSs) usually operate in dynamic, complex, and sometimes unpredictable environments. The uncertainty they face during their operations can stem from various sources, including fluctuating environmental conditions or unexpected system behavior. That uncertainty introduces inherent risks that can compromise the safety and reliability of these systems. The use of safety cases endeavors to address and mitigate such risks, aiming to establish a robust framework that ensures the system's ability to operate safely, even in the face of uncertainty.

Safety cases are usually very large documents that may span several hundred pages [12]. For instance, the safety case of an air traffic control system may consist of more than 500 pages and 400 referenced documents mostly written in natural language [12]. Still, they are usually generated manually, which can be a very tedious, error-prone, and time-consuming process, especially for complex and heterogeneous systems such as cyber-physical systems (CPS) [42, 43, 44]. It is therefore crucial to devise new techniques to (semi-)automatically generate safety cases. In the realm of safety case research, despite the extensive body of work dedicated to automating the generation of (partial) safety cases, our investigation stands –to the best of our knowledge – as a pioneering effort in employing GPT-4 for this purpose. Consequently, the primary focus of our paper revolves around the automation of safety case creation through the utilization of GPT-4. The main objective of this endeavor is to alleviate the workload of safety case modelers, offering a novel approach to safety case development.

## 2.2 How to represent a safety case?

A safety case can be conveyed through various means, including traditional prose, semi-structured text, also known as structured prose, or graphical notations. Initially, safety cases were predominantly presented using traditional prose. However, articulating safety arguments solely in natural language can result in unclear, informal, and occasionally incomprehensible safety cases. An advancement over traditional prose is referred to as structured prose, which allows for the formal representation of relationships between elements within a safety case [45, 46]. In contrast to both plain and structured prose, graphical notations offer a coherent way to depict safety arguments along with their supporting evidence [18].

Structured argument notations, such as the GSN and CAE, offer a means to ensure clarity regarding the inter-dependencies between claims, thereby enhancing the understanding of how one claim relies on another. GSN is one of the most adopted notations used to model assurance cases (e.g., safety cases) [47, 48, 40]. GSN allows depicting an assurance case as a tree-like structure called *goal structure*. The core elements of GSN are goals, solutions, strategies, assumptions, contexts, and justifications [49, 18, 50].

A ***Goal*** is used to articulate a claim. To support evidence, a ***Solution*** is used. To connect goals to sub-goals, an ***Argument Strategy*** is employed. To capture the context of a claim in a goal, a ***Context*** is documented. ***Assumptions*** are statements or propositions that can be accepted without being explicitly proved. They are the foundation upon which the goals and claims are built. ***Justifications*** are needed to provide reasoning or rationale to support the claims and goals. There are two types of relationships that is used between these GSN elements: ***InContextOf*** and ***SupportedBy***. ***InContextOf*** establishes the contextual relationship between the elements. ***SupportedBy*** signifies that one element of GSN provides support or evidence for another element [51, 18].

GSN elements can be decorated by relying on decorators. A few examples of decorators are ***Undeveloped***, which is a decorator indicating that the GSN element is yet to be developed and ***Uninstantiated***, which is a decorator indicating that the GSN element is yet to be instantiated (i.e., the element needs to be replaced by a more concrete instance). When these GSN elements are connected together using the relationships (i.e., SupportedBy and InContextOf), it forms the goal structure [51, 18].

In this paper, our primary focus is on Goal Structuring Notation (GSN). Still, when performing experiments with GPT–4 in previous work (i.e. [52]), we observed that when prompted to generate a safety case in GSN format, GPT–4 produced structured prose instead of graphical goal structures (i.e. GSN diagrams) due to its inability to generate graphical representations. Consequently, we adopted structured prose notation to align with GPT–4's capabilities. Still, the structured prose notation we adopted complies with GSN i.e. uses GSN concepts to textually represent goal structures.

## 2.3 An example of a safety case

Figure 1 is adapted from [4]. It provides an example of a safety case in GSN together with its structured prose equivalent. In that figure, the main objective of the safety case is to show that the map system is acceptably safe to operate, which is depicted by G1. G1 is defined in two contexts, C1, which corresponds to the definition of the map system and C2, which defines the role and context of the map. G1 is further supported by goal G2, which states that all the identified hazards are either eliminated or they have been sufficiently

mitigated for safe operation. G3 is in the context of C3, which states that the hazards in G3 are the hazards identified from DAO (Dysfunctional Analysis Ontology). G2 is further supported by strategy S2 which places an argument over each of the identified hazard. It is also in the context of an Assumption A1 which states that all hazards have been identified. Finally, G3 shows that hazard H1 has been eliminated which is supported by the evidence Sn1, execution of the safety rules. Hence, through a series of goals, strategies, assumption, contexts and solutions, this safety case is able to argue that the map system is sufficiently safe to operate.

G1: Map system is acceptably safe to operate

C1: Map systems is defined

C2: Map role and context

G2: All identified hazards have been eliminated

or sufficiently mitigated

C3: Hazards identified from DAO

S1: Argumentation over each identified hazard

A1: All hazards have been identified

G3: Hazard H1 has been eliminated

Sn1: Safety rules execution

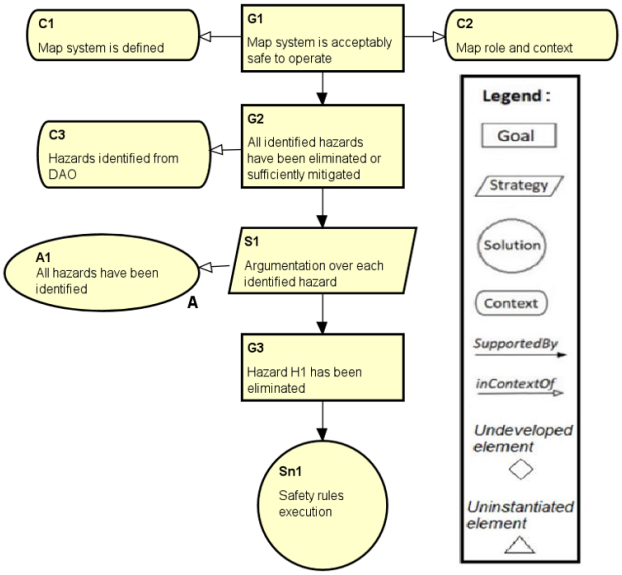


Figure 1: On the right, an example of safety case adapted from [4] and depicted in the GSN; on the left, the equivalent of the safety case in the structured prose.

## 2.4 What is a Large Language Model?

A large language model (LLM) is a type of artificial intelligence (AI) model that have garnered attention in natural language processing tasks (NLP) recently. LLMs are usually transformer models (e.g., GPT-4) that are trained on massive data sets to generate text and answer questions with high accuracy. LLMs are typically characterized by their vast size and complexity, as they are trained on massive data sets that contain text from a variety of sources, including books, articles, websites, and more. These models use deep learning techniques, particularly neural networks, to process and generate text, and they are known for their ability to capture complex patterns and semantics in language [53]. Prominent examples of LLMs include OpenAI's GPT (Generative Pre-trained Transformer) series, such as GPT-3.5 and GPT-4 [30], as well as models from other organizations like BERT (Bidirectional Encoder Representations from Transformers) [54] developed by Google. In this paper, our focus is on GPT-4.0 since it is more powerful than most of the LLMs present in the market and it is relatively more affordable.

# 3 Related Work

## 3.1 Systematic literature reviews on safety cases

In the field of safety assurance, Nair et al. [55] developed a taxonomy for the classifications of information and artifacts considered as safety evidence in various application domains. Tambon et al. [56] identified and analyzed challenges associated with certifying machine learning-based systems and explored proposed solutions found in the literature. Neto et al. [21] provided an overview of the state-of-the-art in the safety assurance of Artificial Intelligence (AI)-based systems and offered guidelines for future work. Mohamad et al. [38], focused on a different type of assurance cases called security cases. They therefore conducted a systematic literature review on Security Assurance Cases to highlight the need for validated methodologies, practical guidance, and dedicated tools to support security case development. Maksimov et al. [12] examined the assessment features implemented in 10 assurance case software tools, discussed their strengths and weaknesses, and identified future research directions.

Bibliometric analysis has been adopted in several different domains including medical, and energy. For example, Han et al. [57] conducted a comprehensive bibliometric analysis of research trends in surgery with mixed reality from 2000 to 2019, revealing key areas of focus such as training tools, clinical applications, etc. Another study by Qin et al. [19] conducted a comprehensive bibliometric review of past green energy adoption (GEA) research and its determinants. It provided insights into authors, countries, journals, and the evolution of GEA research.

None of the aforementioned studies offers an advanced bibliometric synthesis of the existing body of literature on SACs. Most of them mainly provide individual perspectives on specific aspects of safety assurance, such as taxonomy development, AI and machine-learning-based systems, and assurance case tools. Still, they do not comprehensively analyze the entire scientific landscape of safety assurance case research through a bibliometric lens. Our research aims to address this gap by synthesizing collective knowledge within the field of safety cases, providing a holistic perspective that complements the insights gained from these individual studies.

In contrast to the aforementioned studies that narrow their focus to specific facets of safety cases, our quantitative synthesis takes a more comprehensive approach. Rather than concentrating solely on one aspect, we aim to present a holistic perspective by encompassing various types of systems, including but not limited to ML-based systems, such as cyber-physical systems (CPS), adaptive systems, and others.

## 3.2 Assurance Case Generation through Automation

In the last decade, many researchers have actively contributed to the automatic generation of safety cases. For example, Denney et al. [58] propose a methodology that automatically generates safety cases derived from a formal verification method. They have applied their methodology to the Swift Unmanned Aircraft System (UAS) which is being developed at NASA. Nonetheless, the research lacks a method for evaluating or assessing the confidence level of the automatically generated assurance cases. Denney et al. [59] introduce AdvoCATE, an automated tool designed to facilitate the construction and evaluation of safety cases. In addition to manual functions, it offers automated features such as metadata incorporation, translation to

various formats, including compatibility with other safety case tools, composition through auto-generated safety case components, and computation of safety case metrics. The tool primarily aligns with the Goal Structuring Notation (GSN). AdvoCATE possesses the capability to autonomously integrate safety case elements generated manually with content originating from external tools. However, this external integration is limited only to AUTOCERT, a formal verification tool [60].

Hartsell et al. [61] introduce an automated method for constructing assurance cases in the context of Cyber-Physical Systems (CPSs). The method leverages patterns to reduce labor-intensive and error-prone manual efforts in constructing assurance arguments, particularly in CPS development involving interconnected system models. The study provides an illustrative example, showcasing how this approach significantly streamlines the creation of tightly integrated assurance cases with source artifacts. However, the generated assurance case may exhibit incompleteness, as some nodes lack parameters and cannot be automatically instantiated. The extent of automation depends on available patterns and information in the system model set, underscoring the need for additional customization to achieve full coverage.

Similarly, Wang et al. [62], the authors present an automated assurance case generation framework that helps in creating, validating and assessing safety cases. While their work represents a significant contribution to automating safety cases, it is worth noting that in their evaluation, they focus exclusively on specific sections of the assurance cases that are either fully complete or nearly complete. This approach may not provide a completely accurate assessment.

Ramakrishna et al. [63] argue that manual methods of creating safety cases are very brittle and do not have a systematic approach. As an improvement to current practises, the authors introduce a tool named structured ACG (assurance case generation) tool which uses design artifacts, evidence and expertise of developers to construct a safety case automatically. The study's primary limitations are the need for human input in converting certification criteria into GSN goals and in evaluating safety cases with just a single metric, and the lack of automation in linking evidence to system components and defining assumptions.

### 3.3 Approaches focusing on safety argument generation in the automotive domain

The ISO 21448:2022 standard [3] offers a comprehensive framework and guidance for ensuring the safety of the intended functionality (SOTIF). SOTIF aims to prevent unreasonable risks resulting from hazards caused by two main factors: a) deficiencies in specifying the intended functionality at the vehicle level, and b) inadequacies in specifying or performing electric and/or electronic (E/E) elements within the system. It provides direction on the design, verification, and validation measures, along with operational phase activities necessary to achieve and uphold SOTIF. This guidance is applicable to intended functionalities where maintaining a clear situational awareness is critical for safety, and such awareness relies on intricate sensors and processing algorithms

Hawkins et al. [3] discuss the challenges involved in ensuring the safety of integrating ML enabled component into safety-critical systems.To address this challenge, they introduce *"Assurance of Machine Learning for use in Autonomous Systems (AMLAS)"* methodology. AMLAS offers a structured approach, consisting of safety case patterns and a systematic process, aimed at integrating safety assurance into ML component development. Furthermore, it provides a means to generate the necessary evidence for explicitly justifying

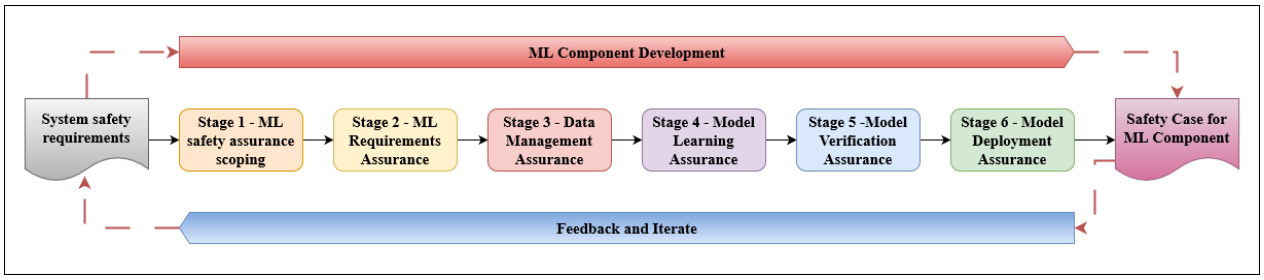


Figure 2: Overview of the AMLAS framework adapted from [3]

the safety of these components when integrated into autonomous system applications.
Figure 2 adapted from [3] shows a high-level overview of the AMLAS process. Throughout the paper, the notation [A]–[HH], refers to the individual artefacts prescribed by AMLAS [2, 3]. The stages of AMLAS are as follows:

1. **Stage 1 - ML assurance scoping** : In this stage the authors define the scope of ML safety assurance and also the scope of the safety case. Finally, they provide the ML assurance scoping argument pattern. This stage has the following set of input and output artefacts. Input artefacts needed are **System Safety Requirements [A], Description of Operating Environment of System [B], System Description [C], ML Component Description [D], ML Assurance Scoping Argument Pattern [F]**. The outputs artefacts generated are **Safety Requirements Allocated to ML Component [E], and ML Safety Assurance Scoping Argument [G]**

2. **Stage 2 - ML requirements assurance** : In the second stage, the authors derive the safety requirements allocated to the ML component based on the system safety requirements. Then, they validate these requirements and provide the ML safety requirements argument pattern. The input artefacts needed are **Safety Requirements Allocated to ML Component [E] and ML Safety Requirements Argument Pattern [I]**. The output artefacts of this stage are **ML Safety Requirements [H], ML Safety Requirements Validation Results [J], ML Safety Requirements Argument [K]**.

3. **Stage 3 - Data management assurance** : In this stage the authors develop the data requirements that align with the ML safety requirements and generate data sets in accordance with those requirements. Provide the ML data argument pattern. The input artefacts in this stage are **ML safety requirements [H] and ML data argument pattern [R]**. The output artefacts are **Data requirements [L], Data requirements justification report [M], Development data [N], Internal test data [O], Verification data [P], Data generation log [Q], ML data validation results [S] and ML data argument[T]**.

4. **Stage 4 - Model learning assurance** : In this stage the authors develop the actual ML model using the development data obtained in the previous stage and use internal test data to assess the model. Finally, the provide the ML learning argument pattern. The input artefacts are **ML safety requirements [H], Development data [N], Internal test data [O], ML learning argument pattern [W]**. The output artefacts are **ML model [V], Internal test results [X], ML learning argument [Y], Model development log [U]**.

5. **Stage 5 - Model verification assurance** : In this stage the authors demonstrate that the developed ML model actually meets the safety requirements and the provide the ML verification argument pattern. The input artefacts in this stage are **ML safety requirements [H], Verification data [P], ML Model [V], ML verification argument pattern [B]**. The output artefacts are **ML verification results [Z], Verification log [AA], ML Verification argument [CC]**.

6. **Stage 6 - Model deployment assurance** In this stage the authors integrate the machine learnt model into the target system and validate that the ML safety requirements are still satisfied even after the model integration into the system. They also provide the ML deployment argument pattern. The input artefacts at this stage are **System safety requirements [A], Environment description [B], System description [C], ML model [V], ML deployment argument pattern [GG], Operational scenarios [EE]**. The output artefacts are **Erroneous behaviour log [DD], Integration testing results [FF], ML deployment argument [HH]**.

In the work by Borg et al. [2], they demonstrate a safety case for an ML component, known as SMIRK, in an open automotive system using AMLAS. SMIRK is a pedestrian automatic emergency braking system operating in an industry-grade simulator, and its ML component relies on a deep neural network for pedestrian recognition. This work represents the first complete application of AMLAS outside of its original authors, showing how to provide sufficient evidence of ML safety within SMIRK's operational design domain (ODD). Despite having a minimalistic ODD, the ML safety case is substantial, emphasizing the need for a thorough explanation of AI engineering aspects. The SMIRK system and its safety case are made available under an open-source license for the research community. Our approach is an extension of AMLAS [3] and SMIRK [2].

Gauerhof et al. [64] elucidate the prevalent use of machine-learned models (MLMs) in contemporary self-driving cars, particularly in tasks such as object detection, including pedestrian detection. They emphasize that the failure of these models can lead to catastrophic consequences, including loss of life and severe injuries. Consequently, they underscore the criticality of ensuring the safety of such MLMs to guarantee the safe operation of self-driving vehicles. The primary contribution of their research lies in demonstrating how safety requirements can be systematically derived and refined throughout various stages of the ML lifecycle process, with a specific emphasis on data management and model learning requirements.

Ashmore et al. [65] provide a comprehensive survey of the state-of-the-art in machine learning (ML) assurance, which involves generating evidence to demonstrate that ML is safe for its intended use. The survey covers various methods capable of providing such evidence at different stages of the ML lifecycle, which encompasses the entire process from data collection for training an ML component to its deployment within a system. The main focus of the research is to define specific safety and reliability criteria called **assurance desiderata** for each stage of the ML lifecycle. They also review the current methods that help meet these criteria and pinpoint areas where further research and development are needed. This approach aims to ensure the safe and effective use of machine learning in critical applications.

### 3.4 Automated Software Modeling with Large Language Models

In recent times, the utilization of Large Language Models (LLMs) has seen a significant rise in various software engineering tasks, ranging from recommending design concepts for metamodel development to generating goal models and Unified Modeling Language (UML) models. This surge reflects the increasing recognition of LLMs' potential to enhance and automate various aspects of the software engineering process. For instance, in the work by Weyssow et al. [66], the authors address the challenge of designing metamodels in Model-Driven Engineering (MDE) and propose an approach utilizing Deep Learning, specifically, pre-trained language models, to recommend relevant domain concepts for metamodel designers. By training the model on a substantial metamodel dataset, incorporating both structural and lexical properties, the study demonstrates the model's accuracy in providing recommendations for concept renaming scenarios.

In a more recent development, Chen et al. [29] explore the capabilities of advanced language models like GPT-4 in requirements engineering, particularly within the realm of goal-oriented models. The research focuses on GPT-4's proficiency in the Goal-oriented Requirement Language (GRL) and finds that GPT-4 possesses substantial knowledge about goal modeling, offering valuable insights, especially for stakeholders less familiar with the domain. Our paper builds upon and extends this research.

Furthermore, Chaaben et al. [67] and Camara et al. [68] have also employed ChatGPT for UML model generation. The former introduces a novel approach for enhancing domain modeling using LLMs without extensive training, leveraging few-shot prompt learning. The latter explores the potential impact of LLMs like ChatGPT on software modeling tasks, specifically in the context of UML modeling, highlighting their capabilities and limitations in this domain.

Viger et al. [69] proposed to use generative AI to elicit defeaters i.e. potential doubts in the arguments. They proposed to use GPT-4 to aid in the identification of defeaters in assurance cases to make them more reliable. That work is still preliminary, so the authors did not empirically validate it.

The study by Ahmad et al. [70] addresses the complexities of architecting software-intensive systems, highlighting the challenges faced in architecture-centric software engineering (ACSE). It introduces the concept of software development bots (DevBots) trained on large language models, such as ChatGPT, to facilitate collaborative ACSE. The research presents a case study involving a novice software architect and ChatGPT to architect a service-based software, demonstrating the potential of AI-driven decision support in this domain.

The case study by Djaber and Ismail [71] utilizes GPT-4 in creating an E-commerce shoe sales platform. The GPT-4 model automates UML diagrams, code, UI prototypes, and database schemas, improving efficiency and consistency. The research highlights AI's transformative role in software engineering, bridging theory and practical implementation for more efficient and high-quality development.

The study by Fill et al. [72] conducted experiments using ChatGPT and the latest GPT-4 models to explore future possibilities for generating and interpreting conceptual models, including ER, Business Process, UML class diagrams, and Heraklit models. The authors' aim is to inspire the community to build upon these examples and explore their own innovative approaches in modeling applications leveraging large language models like ChatGPT.

A common limitation across the previously mentioned references is that none of the mentioned studies lever-

aged generative AI to create safety cases. While they explored various applications of large language models (LLMs) in software engineering tasks such as metamodel design recommendations, goal modeling, and UML model generation, they did not extend their focus to automating the process of generating safety cases. This underscores a potential untapped area where generative AI could play a significant role in enhancing safety assurance in software engineering contexts.

# 4 Methodology

## 4.1 Methods used to answer RQ1

In choosing to conduct a bibliometric review rather than a traditional literature review, we recognized the inherent complexity and vastness of the research landscape surrounding safety assurance cases. With an extensive body of literature spanning various disciplines and domains, a comprehensive literature review would have been a monumental task, requiring substantial time and resources. Instead, by harnessing the power of bibliometric analysis, we aimed to distill key insights and trends from a large corpus of primary studies efficiently. This approach allowed us to quantitatively analyze publication trends, identify influential keywords and primary studies, and gain a holistic understanding of safety case research landscape in a more time-effective manner. Thus, our decision to pursue a bibliometric review was driven by a desire to maximize the depth and breadth of our analysis while optimizing our available resources.
To conduct our bibiliometric analysis, we have followed the guidelines of PRISMA 2020 [73] and SEGRESS [74]. To manage the references of the primary studies included in our analysis, we have used EndNote reference management tool [75]. We primarily used the database-driven search method, a commonly employed technique, to search for primary studies.

### 4.1.1 Information sources

The most commonly utilized search technique is database-driven, and we employed this approach to search for primary studies. Our search focused on five common databases: IEEE Explore [76], ACM Digital Library [77], SCOPUS [78], Engineering Village (EV) [79], and Google Scholar [80]. For our search on Google Scholar, we relied on the Publish Or Perish tool developed by Harzing [81].

### 4.1.2 Identification strategy

In order to identify studies within the databases mentioned in the previous section, we initiated our search with a preliminary query using the terms ”Safety Case” OR ”Safety Assurance Case.” This initial query yielded 865 references. However, upon closer examination, it became evident that some relevant papers were missing from this initial search. To address that gap, we adopted an iterative and incremental approach, systematically enhancing our query by incorporating additional keywords derived from an extensive review of the titles and abstracts of these papers.
Furthermore, we also incorporated keywords (e.g., claim, argument, evidence, justification, assurance) from the search queries employed by Maksimov et al. [12] and Mohamad et al. [38] to ensure comprehensive coverage. To ensure the queries used in the database-driven search effectively capture a substantial yet manageable body of relevant literature, two researchers (i.e. a graduate student and a faculty member) from our team meticulously refined the queries multiple times. A third one (i.e., a faculty member) reviewed the query and provided recommendations to make sure it is inclusive enough. Eventually, we refined our search queries to the following:

**QUERY-1:** *"Safety Claim" OR "Safety argument" OR "Safety evidence" OR "Safety justification" OR "Safety case" OR "Safety assurance case" OR "Safety assurance" OR "Safety compliance" OR "Safety artifacts"*

**QUERY-2:** *("Safety Case") AND ("GSN" OR "CAE" OR "SACM")*

When conducting the database-driven search, we executed the above two search strings in the titles, abstracts, keywords, or metadata of the studies available in the five databases (see Section 4.1.1).

### 4.1.3 Specification of eligibility criteria

Table 1 reports the eligibility criteria we used to select the primary studies we included in our literature review.

Table 1: Inclusion and exclusion criteria

| Inclusion criteria | Exclusion criteria |
|---|---|
| 1. Peer-reviewed Conference, Journals, Workshops papers<br>2. Papers that present an approach to manage (e.g., create, assess, reuse, maintain) safety cases<br>3. Publication year between 2012 and 2022 | 1. Books, Posters, Book chapters, Theses, Tutorials, Secondary studies<br>Papers that focus on general assurance and safety rather than safety cases<br>3. Non-English publications<br>4. Short studies less than 4 pages<br>5. Studies inaccessible due to paywalls<br>6. Unsuccessful attempts to secure free access from authors<br>7. Studies that do not propose techniques focusing on safety case |

### 4.1.4 Selection strategy

To select and filter out primary studies according to the eligibility criteria listed above, we employed a five-phase process explained as follows:

- ***Phase-1:*** In this phase, we gathered the records collected from all 5 databases (using both queries) and import them to EndNote, a reference management tool [75, 82].
- ***Phase-2:*** In this phase, we harnessed the duplicate records detection feature of EndNote tool to eliminate duplicates.
- ***Phase-3:*** From the third phase, we manually filtered the records, starting with scanning the records titles based on the inclusion and exclusion criteria
- ***Phase-4:*** In the fourth phase, we scanned the abstracts and titles of the records based on the inclusion and exclusion criteria.
- ***Phase-5:*** Finally, we scanned the entire text of the document based on the inclusion and exclusion criteria. This allowed us to get the final list of primary studies on which our literature review focuses.

#### 4.1.5 Data extraction

To collect data from the primary studies, we employed structured data extraction forms designed as Excel sheets. We used these forms to record bibliometric information (e.g., authors, titles, and publication venues) for all the studies incorporated into our study.

#### 4.1.6 Bibliometric synthesis

The adoption of bibliometric synthesis is on the rise across various research fields ([83, 84, 85, 86, 87]). To complete our bibliometric synthesis, we employ the widely used bibliometric tool VosViewer [88] to extract and analyze bibliometric data from primary studies. This tool facilitates the automated creation of charts presenting the bibliometric characteristics of the primary studies. Additionally, for generating supplementary charts, we utilize Google Charts [89], Microsoft Excel, and leverage Tableau [90]. The latter is a prominent data visualization tool extensively used in the data analytics sector.

### 4.2 Methods used to answer RQ2

To answer RQ2, we have chosen to extend the SMIRK [2] and AMLAS [3] methodologies for a specific reason. AMLAS represents a significant milestone as the first methodology to outline the process of constructing a safety case for a machine learning component integrated into autonomous vehicles or advanced driver assistance systems (ADAS) [3]. Prior to AMLAS, industry standards like ISO 26262 or ISO 21448 provided guidelines for automated driving and automotive safety but did not explicitly detail the specific activities or evidence required when integrating an ML component into autonomous vehicles [64].
Similarly, SMIRK represents a pioneering research paper that actively applied the AMLAS methodology to an ML component focused on pedestrian detection in an ADAS context [2]. What sets SMIRK apart is its comprehensive demonstration of a complete safety case, complete with safety arguments for each phase of the AMLAS framework and the creation of 34 safety artefacts [2]. In essence, SMIRK acts as a valuable starting point and engineering research reference. It simplifies the process for researchers like us to construct a safety case for an ML-enabled component within autonomous vehicles [2]. This extension allows us to adapt these methodologies to our specific requirements while building upon the foundational work provided by AMLAS and SMIRK, streamlining the development of safety cases in the context of machine learning integration.
In our research, we introduce a framework designed to establish a comprehensive safety case for a Machine Learning component integrated into an autonomous vehicle. Figure 3 depicts the overview of our methodology. It is important to note that, unlike SMIRK and AMLAS, our approach deviates slightly as it focuses on a scenario where the ML component is already developed, trained, tested, and operational within the AV. Our primary objective is to create a thorough safety case for the existing system, emphasising the critical goal of demonstrating that the ML component running within the AV ensures an acceptable level of safety and effectively prevents collisions.
Table 2 reports the key differences between our design approach and existing design approaches (e.g., AMLAS and SMIRK). We further elaborate on our design approach in the remainder of this section.

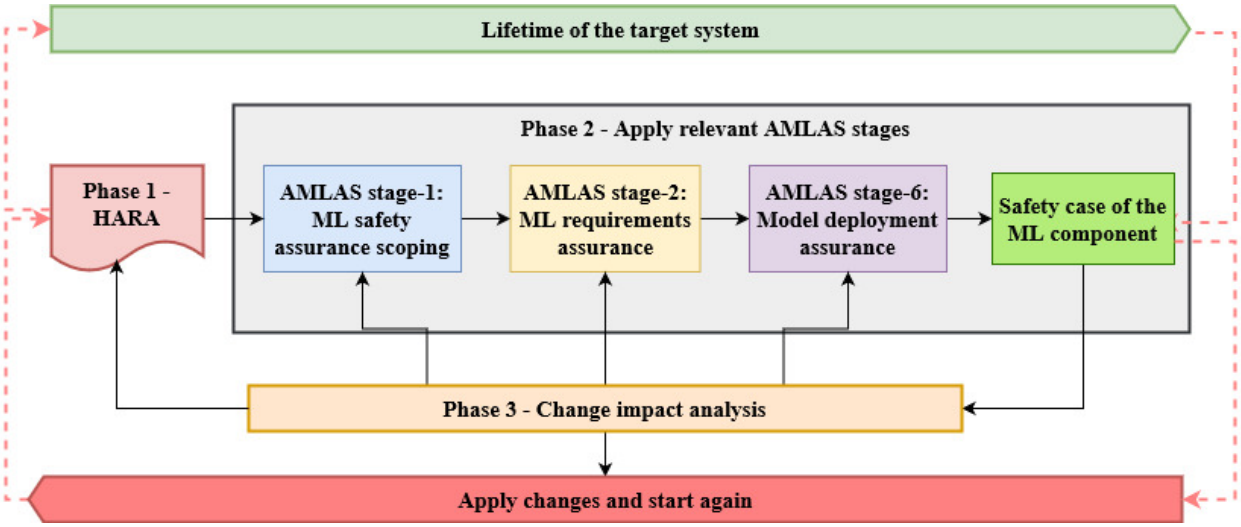


Figure 3: Overview of our safety case creation approach based on [2] and [3]

| Difference | SMIRK | AMLAS | Ours |
|---|---|---|---|
| Stages | Same as AMLAS | 6 stages | 3 stages plus an additional change impact analysis phase |
| ML component | Pedestrian detection | - | Reinforcement learning algorithm for avoiding collisions |
| System | ADAS Simulation [2] | - | Quanser Qcar [91] |
| System status | Developed concurrently while applying AMLAS | Provides guidelines for concurrent development of ML component and safety case | ML development already completed |

Table 2: Key differences between SMIRK [2], AMLAS [3] and our framework

### 4.2.1 Phase 1: Hazard Analysis and Risk Assessment

According to AMLAS, the input to the first stage is the system safety requirements. There are many methodologies to derive the safety requirements. For example, the requirements can be taken from the software requirement specification (SRS) document or methodologies like HARA or HAZOP can be performed to identify the hazards and safety requirements. In our work, we chose to perform HARA. The reason being, ISO 26262-3:2018 standard mandates that automotive systems undergo a hazard analysis and risk assessment (HARA) to identify the required safety measures for implementing a specific feature [5]. According to ISO 26262-3:2018, it is *"a method to identify and categorize hazardous events of items and to specify safety goals and ASILs related to the prevention or mitigation of the associated hazards in order to avoid unreasonable risk"* [5]. Therefore, we initiate the safety argumentation process with HARA [5] to identify hazards that have the potential to lead to system failures. The identification of these hazards serves as a pivotal foundation for shaping the overall system requirements. These requirements, in turn, act as a basis for deriving the prerequisites for subsequent stages within the AMLAS framework [2]. Furthermore, these established requirements also serve as essential objectives within the safety cases [92]. Demonstrating that these goals have been successfully met is a key indicator of the overall system's safety and reliability. In essence, this initial step is crucial in our approach to developing a robust safety case for the system. To perform HARA, we follow the guidelines presented in ISO 26262-3:2018 [5] together with the methodology that Beckers et al. [92] introduced.

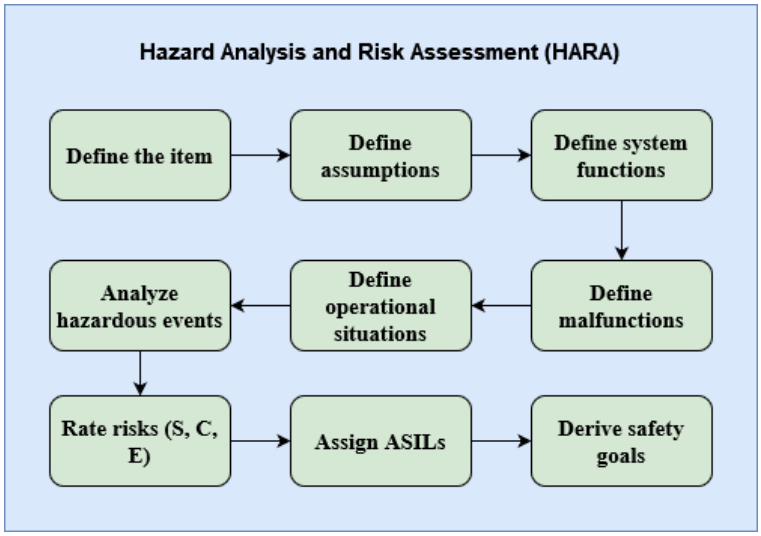


Figure 4: Overview of hazard analysis and risk assessment (HARA) process adapted from [5].

Part 3 of ISO 26262 involves identifying potential hazards by analyzing the operational situations of the item, which could be a vehicle, a vehicle system, or a vehicle function [5, 93]. These potential hazards are categorized based on severity, probability of exposure, and controllability. Subsequently, an Automotive Safety Integrity Level (ASIL) is assigned to each potential hazard [5, 93]. The ASIL is also allocated to the safety goal(s) designed to prevent or mitigate the potential hazard, ensuring unreasonable risk is avoided. Safety requirements, or risk reduction (safety) requirements, are then derived from these safety goals and

inherit the corresponding ASIL [5, 93]. The following steps provide a brief overview of the processes happening at this phase. We adopted these steps from [5, 92, 93]. Figure 4 provides an overview of the steps in HARA.

- **Step 1 - define the item:** Before starting the hazard analysis, we first need to define the item (system). Here, we define the item for which we are developing the safety case, its functionalities and the operating environment. For this purpose, we have used the functional block diagram.
- **Step 2 - Define the assumptions:** In this step, we define the assumptions present in the system very clearly. The assumptions have have four states: accepted, rejected, on hold, ongoing.
- **Step 3 - Define the system functions:** In this step, we provide a short description of the features of the target system.
- **Step 4 - Define malfunctions:** In this step, we define the possible malfunctions that may occur in the lifetime of the target system. For the purpose of defining the malfunctions, we make use of the fault-type guide words adopted from HAZOP, similar to [92]. These keywords include too much,less, too fast, unintended, etc. Finally, we derive the malfunctions by combining the system functions (expected behaviour) with the fault-type guide words.
- **Step 5 - Define operational scenarios:** In this step, we define the different characteristics of the operational scenarios. [5] defines operational scenarios as *"scenario that can occur during a vehicle's life"*. They may be a combination of different aspects such as road type, weather, speed of the vehicle, time of the day, elements on the road etc.
- **Step 6 - Analyze the hazardous events:** In this step, we analyse the hazardous events. [5] defines hazardous event as a *"combination of hazard and an operational situation"*. Through the combination of function, malfunction and operational scenario (FMOS), we identify these hazardous events. We begin with systematically analyzing all possible combinations of FMOS. We also summarize the effects of these identified hazardous events.
- **Step 7 - Rate the risks by severity, exposure and controllability (S,E,C):** After identifying hazards in step 3, we then proceed to classify these hazards based on their severity, exposure and controllability. Figure 5 lists the classes of each of the severity, exposure and controllability levels. For this purpose, we use the following notations. Severity, denoted as S, encompasses four classes ranging from S0, indicating no injuries, to S3, representing fatal injuries. Exposure, labeled as E, ranges from E0, indicating an extremely unusual situation, to E4, indicating a highly likely situation. Lastly, controllability, designated as C, ranges from C0, signifying simply controllable, to C3, indicating uncontrollable.
- **Step 8 - Determine automatic safety integrity level (ASIL):** For each hazardous event identified, we determine the ASIL based on the parameters S, E, and C, defined in the previous step. There are four ASIL levels ranging from ASIL A, denoting the lowest safety integrity level, to ASIL D, representing the highest level. Figure 6 provides the mapping of ASIL levels to the parameters S, E, and C, as defined in the ISO 26262-3:2018 standard.

- **Step 9 - Define safety goals:** For each hazardous event assessed in the hazard analysis, we establish a safety goal. Functional safety requirements necessary to mitigate unreasonable risks for each potential hazard are then derived from these safety goals, which are framed in terms of functional objectives rather than technological solutions. The ASIL of the safety goal is inherited by the functional safety requirements derived from it. The ASIL assigned to the hazardous event is also allocated to the corresponding safety goal. It's possible for a potential hazard to have multiple safety goals, and if similar safety goals are identified, they can be consolidated into a single safety goal assigned the highest ASIL among them. Hazardous event with ASIL QM need not be converted into safety goal [5].

At the conclusion of HARA, we will have established the safety goals that will form the basis of the safety requirements that we present in the subsequent phases.

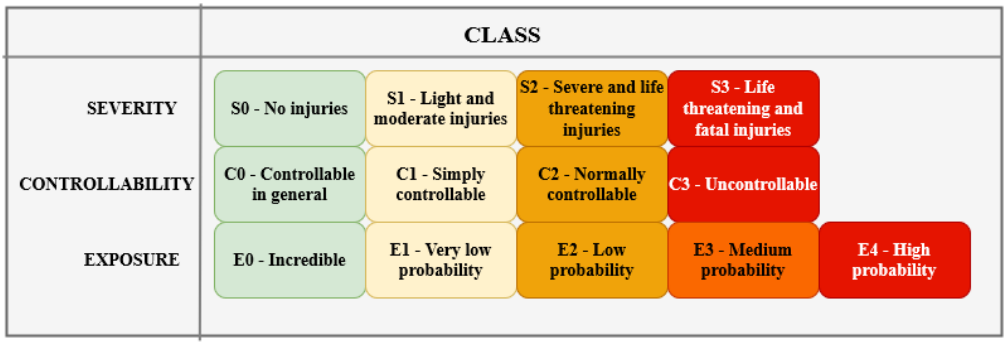


Figure 5: Description of each of the classess of severity, exposure and controllability as explained in [5]

| Severity class | Exposure class | Controllability class | | |
|---|---|---|---|---|
| | | C1 | C2 | C3 |
| S1 | E1 | QM | QM | QM |
| | E2 | QM | QM | QM |
| | E3 | QM | QM | A |
| | E4 | QM | A | B |
| S2 | E1 | QM | QM | QM |
| | E2 | QM | QM | A |
| | E3 | QM | A | B |
| | E4 | A | B | C |
| S3 | E1 | QM | QM | A |
| | E2 | QM | A | B |
| | E3 | A | B | C |
| | E4 | B | C | D |

Figure 6: Mapping of ASIL levels to the parameters severity, exposure and controllability adapted from [5]

### 4.2.2 Phase 2: Application of relevant AMLAS stages

Following the identification of hazards in the first phase, we proceed to the second phase where we implement the AMLAS methodology, drawing inspiration from SMIRK [2] but tailoring it to suit the specific

characteristics of our system. Following an analysis of the AMLAS methodology, we have opted to incorporate only stages 1, 2, and 6 into our methodology, as these stages align with our completed system. Stages 3, 4, and 5, which involve data generation, model development, and model learning processes, respectively, have already been finalized in our case. Therefore, it is unnecessary to include these stages in our methodology as they do not pertain to our current objectives. At the end of this phase, we will have a complete safety case (along with artefacts).

#### 4.2.3 Phase 3: Change impact analysis

In the AMLAS methodology, the authors concurrently apply the AMLAS stages alongside the machine learning development lifecycle. Similarly, in the SMIRK approach, the authors initiate AMLAS as soon as the development of SMIRK commences. However, a significant distinction exists between AMLAS, SMIRK, and our approach. In our case, the primary objective is to construct a safety case for a reinforcement learning (RL) algorithm already operational on a self-driving vehicle. To integrate AMLAS into our framework, we must systematically select the relevant stages that apply to our situation. To address this difference, we have added a third phase where we have added change impact analysis, as depicted in Figure 3. We position the change impact analysis as the third phase because certain components of the RL algorithm or the system itself might undergo modifications. These alterations could potentially impact the components of the safety case. Therefore, it is essential for us to conduct a change impact analysis to identify any changes affecting the safety case components. Subsequently, based on the analysis results, we can re-run the stages where changes have an impact and update the safety case accordingly. This iterative process ensures the ongoing validity and effectiveness of the safety case in light of any system or algorithmic modifications. In this work, we implement phase 1 and phase 2 of our framework.

### 4.3 Methods used to answer RQ3.1 and RQ 3.2

#### 4.3.1 High-level overview of our methodology

Figure 17 provides a high-level overview of our methodology. We further describe that methodology in the reminder of this section.

#### 4.3.2 GPT-4 setting

To address our research questions (RQs), we have utilized the online web platform for ChatGPT, specifically the premium version of GPT-4, available at `https://chat.openai.com/`. This platform is instrumental in running our prompts and retrieving GPT-4 answers.

#### 4.3.3 How to prompt GPT-4 to answer RQ3.1?

**Extraction of structural and syntactic rules from GSN standard v.3** To document the structural and syntactic rules governing the Goal Structuring Notation (GSN), we referenced the GSN standard v.3 document [49]. That document has been created and is maintained by the assurance case working group (ACWG),

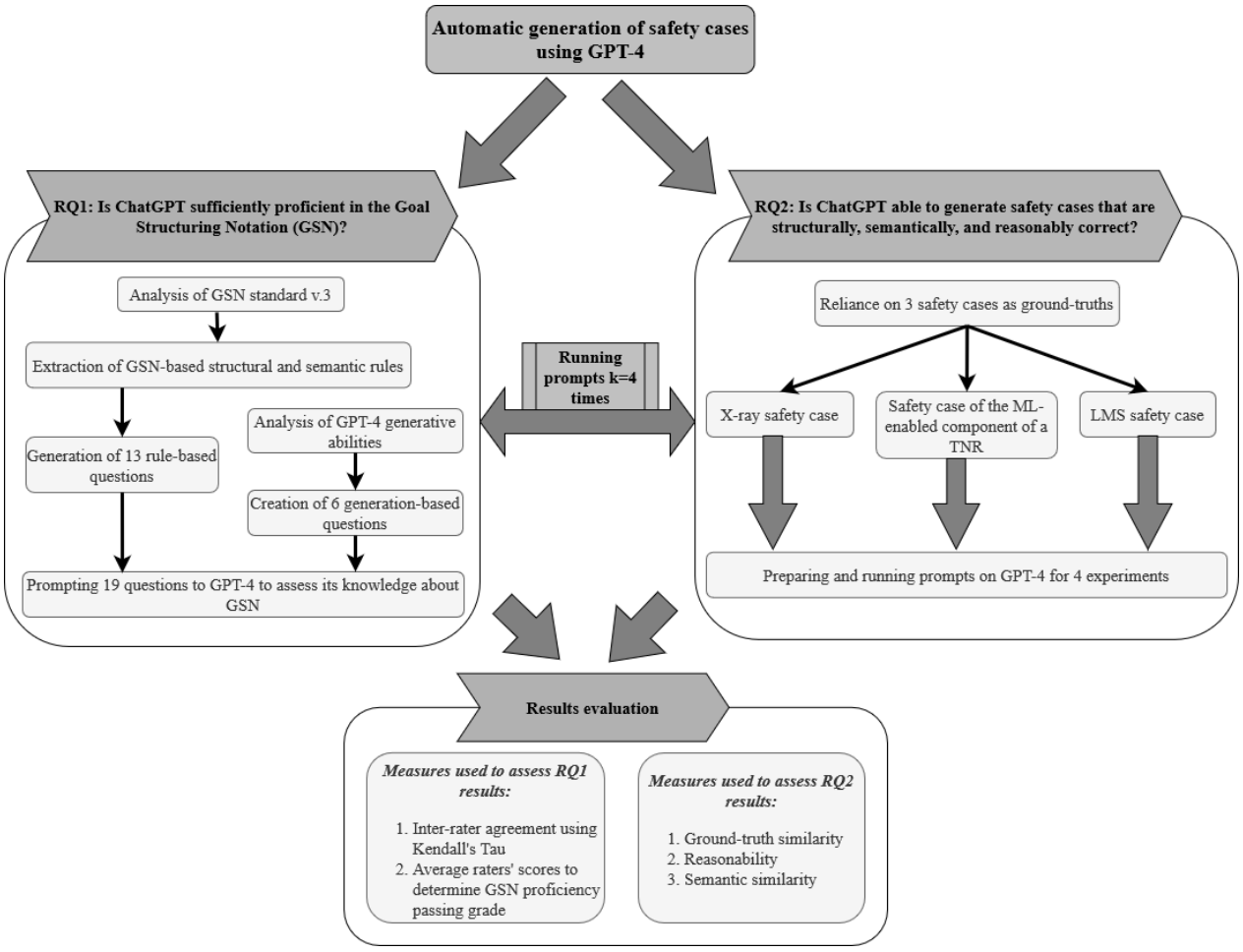


Figure 7: High-level overview of our methodology

with the objective of disseminating information and resources related to GSN [49]. We systematically analyzed that document to extract and derive the structural and syntactic rules that the GSN embodies. This analytical process involved a detailed examination of the standard's content. The rule extraction plays a pivotal role in our research as it provides the essential groundwork for formulating fundamental GSN-related questions. Table 3 outlines these extracted rules. Structural rules pertain to the arrangement and configuration of GSN elements, specifying how each element should be structured and the connections they should have with other elements.

The rules that Table 3 reports define the hierarchy and relationships within GSN, ensuring that the notation is used consistently and effectively. On the other hand, syntactic rules focus on the textual content and meaning contained within GSN elements. These rules encompass aspects such as grammar, terminology, and the interpretation of the textual information specified in each GSN element, ensuring that the notation conveys information accurately and unambiguously.

**Description of the questions asked to GPT-4** Table 4 presents the questions which we framed, for evaluating GPT-4's proficiency on GSN. As detailed in the preceding section, we devised a set of 19 questions,

Table 3: GSN structural and syntactic rules

| GSN element | Allowed connections | Syntactic rules |
|---|---|---|
| Goal | Goal to goal, Goal to strategy, Goal to solution, Goal to context, Goal to assumption, Goal to justification | Noun phrase + verb phrase |
| Context | Goal to context, Strategy to context | Noun phrase |
| Strategy | Strategy to goal, Strategy to context, Strategy to assumption, Strategy to justification | Brief description of the argument approach |
| Solution | Goal to solution | Noun phrase |
| Justification | Goal to justification, strategy to justification | Noun phrase + verb phrase |
| Assumption | Goal to assumption, strategy to assumption | Noun phrase + verb phrase |

similar to [29], to gauge GPT-4's competence in responding to inquiries related to Goal Structuring Notation (GSN), drawing inspiration from the GSN standard v.3 document [49]. Our assessment encompasses the following two categories of questions.

1. ***Rule-based questions:*** we created an initial set of 13 questions, each addressing either the structural or syntactic rules conveyed by the GSN standard
2. ***Generation-based questions:*** In addition to the initial set of 13 questions, we introduced 6 generation-based questions to evaluate GPT-4's capability to generate fundamental GSN elements. These questions assess GPT-4's aptitude in generating core components of a goal structure.

**Description of the prompt structures used for RQ3.1** Before asking the 19 questions to GPT-4, we took a preparatory step by employing specific prompting techniques outlined in [94] and [95]. The primary objective of this preparatory prompt (presented in the text box below) was to orient GPT-4 appropriately, ensuring that it understood its role as an assistant in helping us address inquiries regarding GSN. Furthermore, we considered the tendency of GPT-4 to provide lengthy responses. To mitigate this, we included a directive statement requesting GPT-4 to provide concise answers. This was crucial to maintain the clarity and brevity of the responses. Additionally, we took into account that, without explicitly mentioning "GSN" as an abbreviation for goal structuring notation, GPT-4 might interpret it differently, such as "*Game Show Network*". To avoid ambiguity and ensure precision, we followed the approach established by [29], where setting a context becomes essential. This context-setting step ensured that GPT-4 would respond accurately and in alignment with our intended subject matter, which is GSN. The preparatory prompt is given below:

> You are an assistant that helps me answer questions about Goal structuring notation (GSN). GSN always refers to Goal Structuring Notation from this point. Your answers should be concise and to the point. It should not be more than 2-3 lines

#### 4.3.4 How to prompt GPT-4 to answer RQ3.2?

**Ground-Truth Safety Case Descriptions** To assess both the structural accuracy and syntactic correctness of GPT-4 generated safety cases, we will utilize well-established reference safety cases as comparative

Table 4: List of all 19 questions framed by extracting structural and syntactic rules from GSN standard v.3

| No. | Type | Question |
|---|---|---|
| Q1 | Structural-rule based | What is GSN ? |
| Q2 | | What type of graph is a goal-structure in GSN ? |
| Q3 | | How many elements are present in a goal-structure and what are they ? Can a parent element have multiple children? |
| Q4 | | Can you explain how each of the six elements of GSN are structurally represented. i.e., what shapes? |
| Q5 | | Specify the allowed connections of the all the six elements |
| Q6 | | List all the decorators in GSN along with their representations. Explain off-diagram and uninstantiated decorator |
| Q7 | | Explain the core GSN relationships, their representations and allowed connections ? |
| Q8 | Syntactic-rule based | Explain the syntactic structure of the core GSN elements. i.e., whether they can have a noun or a verb or both. |
| Q9 | | Give some context on the textual elements of Core GSN elements. Can they be stated as a paragraph or should they be stated atomically ? |
| Q10 | | Explain what a top-level claim is. Can it contain multiple claims ? |
| Q11 | | If an argument contains multiple claims, how should the corresponding GSN elements be structured ? |
| Q12 | | What is considered as a ”DO NOT” when it comes to the textual elements of GSN |
| Q13 | | Can strategies restate or redefine what is in the goal ? Can the strategy be avoided in such cases ? When should a strategy be inserted ? |
| Q14 | Generation-based | Give me a sample goal element connected to 2 sub-goals. |
| Q15 | | Give a sample goal connected to a context, justification and assumption |
| Q16 | | Give me a sample goal connected to a strategy |
| Q17 | | Give me a sample goal element connected to a solution |
| Q18 | | Give me a sample GSN model with all the core GSN elements |
| Q19 | | Give me a sample GSN model with all the core GSN elements and incorporate some decorators |

benchmarks i.e. as ground-truths. These reference safety cases pertain to the energy and automotive application domains and encompass: 1) the X-ray safety case detailed in [6]; 2) the safety case pertaining to a Machine-Learning (ML) enabled tire noise recognition (TNR) component in a vehicle, as documented in [7]; and 3) the safety case of a Lane Management System (LMS) from the automotive domain [8].

**X-ray safety case** The X-ray system described in [6] shares similarities with airport X-ray back-scattering machines and qualifies as a safety-critical system due to the potential harm posed to individuals in proximity if the emitted radiation exceeds acceptable limits.

G1: All causes of overradiation have been eliminated

C1: Identified all cause of overradiation

S1: Argument over all identified causes

G2: Software operating as intended

Sn1: Formal verification Results

Sn2: Testing results

G3: Timer interlock operating correctly

Sn3: Fault Tree analysis

Sn4: Testing results

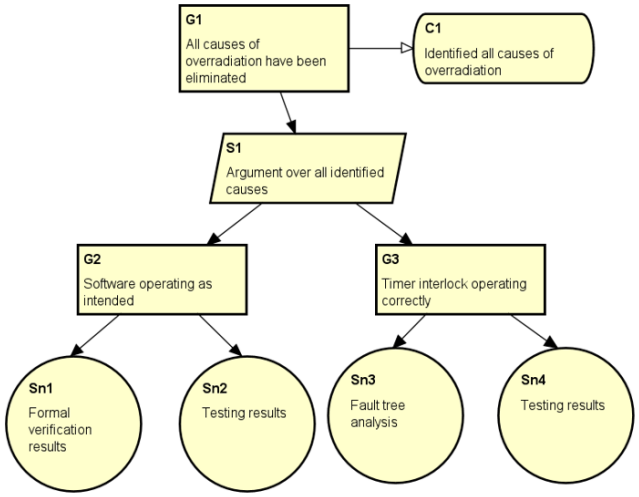


Figure 8: On the right, the X-ray safety case adapted from [6] and depicted in the GSN; on the left, the equivalent of the safety case in the structured prose complying with the GSN.

In Figure 8, we provide a visual representation of the X-ray safety case in both GSN and its equivalent in the structured prose complying with the GSN. The primary objective of the X-ray safety case is to demonstrate the elimination of all factors leading to overradiation (G1). This is substantiated by a strategy (S1) and two subsidiary goals (G2 and G3), along with supporting pieces of evidence (i.e. Sn1, Sn2, Sn3, Sn4), all working together to validate the overarching claim.

**ML safety case** In [7], the authors introduce an assurance case methodology for a Tire Noise Recognition (TNR) component utilizing machine learning (ML) to classify road conditions based on audio signals captured by wheel-mounted microphones. This ML algorithm plays a critical role in safety by providing input to chassis control and powertrain systems, allowing them to adjust control parameters for consistent traction. Real-time processing is essential for accurate road surface assessment, especially in critical situations.
Figure 9 presents a similar representation of the ML algorithm safety case in both GSN and its associated structured prose. The primary aim of the ML algorithm safety case is to confirm that the algorithm complies with the safety requirements allocated to its function (G1). This top goal is reinforced by a strategy (S1), which is further subdivided into two sub-goals (G2 and G3) and substantiated by seven pieces of evidence (Sn1-Sn7).

G1: The machine learning algorithm is intrinsically capable of meeting the safety requirements allocated on the function

A1: Assumptions in safety concept

C1: Technical systems context

C2: Technical safety requirements

S1: Strengths and weaknesses of the algorithm are systematically analyzed

G2: The machine learning approach is well suited to the tasks

Sn1: Description of ML approach

Sn2: Review of generated prototypes

Sn3: Evidence of robustness due to continuous function

G3: Weakness in the approach are mitigated or shown to have no relevant impact on the performance

Sn4: Uncertainty qualification

Sn5: Evidence that prototype adaptation does skew to !dry

Sn6: Maximum distance threshold

Sn7: Out of distribution detection

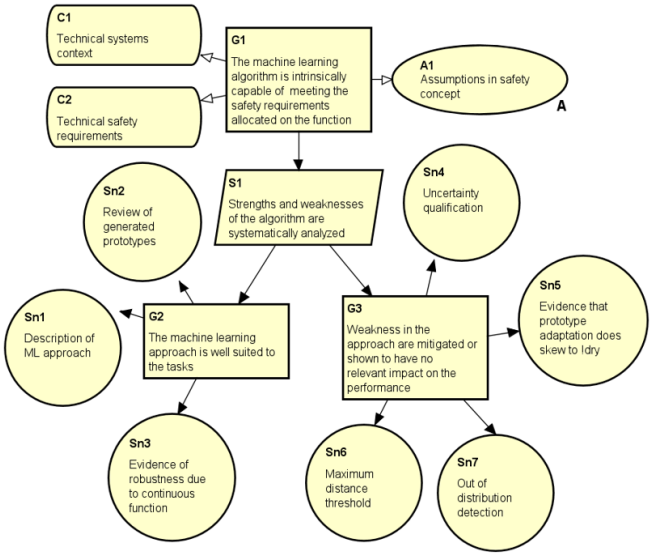


Figure 9: On the right, the ML safety case adapted from [7] and depicted in the GSN; on the left, the equivalent of the safety case in the structured prose complying with the GSN.

**LMS safety case** LMS system defined in [8] is critical in automobiles because they serve as a vital safety feature. Their primary objective is to help maintain the driver's vehicle within or close to the center of their lane. This is of utmost importance as distracted or inattentive drivers can inadvertently drift out of their designated lane, increasing the risk of accidents and collisions [8]. LMS systems mitigate this danger by actively assisting the driver in staying within their lane, thereby enhancing road safety and reducing the likelihood of accidents caused by lane deviation, which can have severe consequences for both the driver and other road users.

Figure 10 presents a similar representation of the LMS safety case in GSN format. The combination of Figures 11, 12 and 13 represent the same safety case in structured prose format complying with the GSN. The main objective of this safety case is to show that the LMS system safety goals are satisfied. Through a series of strategies, sub-goals and solutions, this safety case demonstrates that such a goal is achieved.

**Description of experiments ran to answer RQ3.2** To evaluate GPT-4's ability to generate concrete safety cases, we will adapt the experiments that [29] performed on goal models. For each of the three established safety cases (considered as ground-truths), we will carry out four experiments. In these experiments, we will task ChatGPT (specifically, GPT-4) with automatically generating the same safety cases. The four experiments are explained as follows:

- ***Experiment 1:*** We provide instructions to generate the safety case but provide no domain knowledge, and we do not specify GSN syntax (i.e., no GSN structural rule is specified, and no example of GSN syntax is provided)
- ***Experiment 2:*** We provide instructions to generate the safety case, and we provide domain knowledge, and we do not specify GSN syntax (i.e., no GSN structural rule is specified, and no example of GSN syntax is provided).

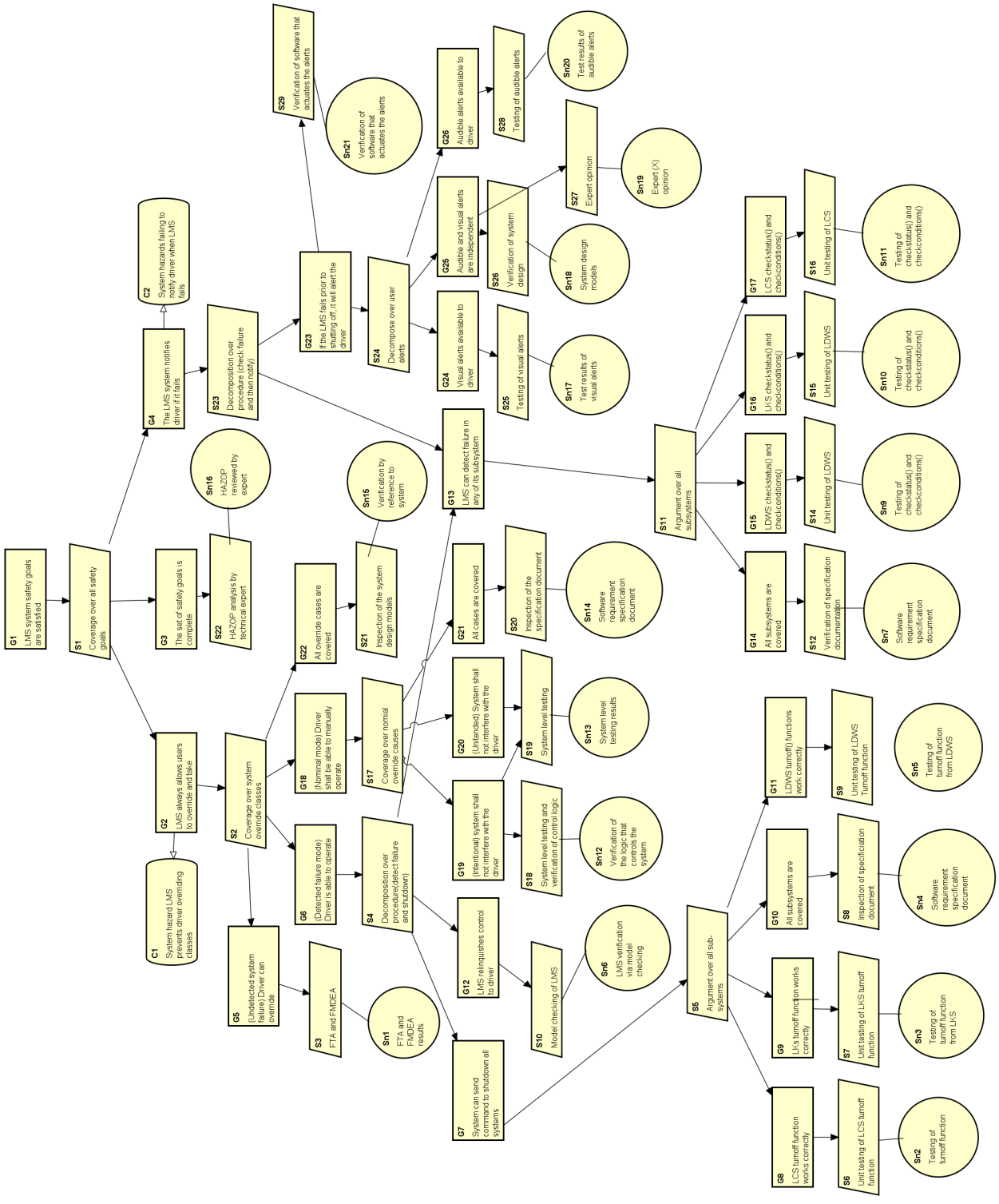


Figure 10: LMS safety case adapted from [8] and depicted in the GSN

G1:LMS system safety goals are satisfied

S1:Coverage over all safety goals

G2:LMS always allows users to override and take

C1:System hazard LMS prevents driver overriding classes

S3:Coverage over system override classes

G5:(Undetected system failure) Driver can override

S4:FTA and FMDEA

Sn1:FTA and FMDEA results

G6:(Detected failure mode) Driver is able to operate

S5:Decomposition over procedure(detect failure and shutdown)

G9:System can send command to shutdown all systems

S5:Argument over all sub-systems

G13:LCS turnoff function works correctly

S6:Unit testing of LCS turnoff function

Sn2:Testing of turnoff function

G14:LKs turnoff function works correctly

S7:Unit testing of LKS turnoff function

Sn3:Testing of turnoff function from LKS

G15:All subsystems are covered

S8:Inspection of specification document

Sn4:Software requirement specification document

G16:LDWS turnoff() functions work correctly

S9:Unit testing of LDWS Turnoff function

Sn6:Testing of turnoff function from LDWS

G10:LMS relinquishes control to driver

S10:Model checking of LMS

Figure 11: LMS safety case adapted from [8] and depicted in the structured prose complying with the GSN - Part-1

Sn7:LMS verification via model checking

G13:LMS can detect failure in any of its subsystem

S13:Argument over all subsystems

G14:All subsystems are covered

S14:Verification of specification documentation

Sn9:Software requirement specification document

G15:LDWS checkstatus() and checkconditions()

S15:Unit testing of LDWS

Sn10:Testing of checkstatus() and checkconditions()

G16:LKS checkstatus() and checkconditions()

S16:Unit testing of LDWS

Sn11:Testing of checkstatus() and checkconditions()

G17:LCS checkstatus() and checkconditions()

S17:Unit testing of LCS

Sn13:Testing of checkstatus() and checkconditions()

G7:(Nominal mode) Driver shall be able to manually operate

S11:Coverage over nominal override causes

G11:(Intentional) system shall not interfere with the driver

S12:System level testing and verification of control logic

Sn8:Verification of the logic that controls the system

G18:(Unintended) System shall not interfere with the driver

S18:System level testing

Sn14:System level testing results

G12:All cases are covered

S19:Inspection of the specification document

Sn15:Software requirement specification document

Figure 12: LMS safety case adapted from [8] and depicted in the structured prose complying with the GSN - Part-2

G8:All override cases are covered

S20:Inspection of the system design models

Sn20:Verification by reference to system

G3:The set of safety goals is complete

S21:HAZOP analysis by technical expert

Sn21:HAZOP reviewed by expert

G4:The LMS system notifies driver if it fails

C2:System hazards failing to notify driver when LMS fails

S22:Decomposition over procedure (check failure and then notify)

G19:If the LMS fails prior to shutting off, it will alert the driver

S23:Decompose over user alerts

G20:Visual alerts available to driver

S24:Testing of visual alerts

Sn22:Test results of visual alerts

G21:Audible and visual alerts are independent

S25:Verification of system design

Sn23:System design models

S26:Expert opinion

Sn24:Expert (X) opinion

G22:Audible alerts available to driver

S27:Testing of audible alerts

Sn25:Test results of audible alerts

S28:Verification of software that actuates the alerts

Sn26:Verification of that actuates the alerts

Figure 13: LMS safety case adapted from [8] and depicted in the structured prose complying with the GSN- Part-3

- ***Experiment 3:*** We provide instructions to generate the safety case, and we do not provide domain knowledge, but we specify GSN syntax (GSN structural rules are specified, and an example of GSN syntax is provided).
- ***Experiment 4:*** We provide instructions to generate the safety case, and we provide domain knowledge, and we specify GSN syntax (GSN structural rules are specified, and an example of GSN syntax is provided).

We ran each of these four experiments ***k*** times, choosing k value as 4, following a similar approach as reported by Chen et al. ([24]). This choice of conducting multiple rounds is motivated by the non-deterministic nature of GPT-4. It helps mitigate potential variability in responses and ensures a more robust and comprehensive evaluation of GPT-4.

**Description of the prompt templates used for RQ3.2** In our approach to framing the prompting templates for RQ2, similar to RQ1, we initiated the process with a preparatory prompt to inform GPT-4 of its role as a safety case developer assistant. Subsequently, for all four experiments, which encompass scenarios with and without context, and with and without GSN syntax, we adopted the prompting techniques outlined in [94] as our foundation. In the referenced paper, the authors offer prompts structured in a question-and-answer format to provide additional context to GPT-4. Building upon this methodology, we tailored our prompts to suit our platform's requirements. These prompts included sets of questions and corresponding answers designed to elucidate the concept of a safety case and the specific format in which we expected GPT-4 to generate the safety case. For experiments involving additional domain context, we extended this approach by incorporating further information in the form of question-and-answer pairs. An example of the prompt template utilized for Experiment-2 (lacking GSN syntax but providing context) in the context of the X-ray safety case (the one in Figure 8) is provided in the text box below:

**Prompt:**
You are a professional safety case developer assistant. I want you to create a safety case for the given system in Goal Structuring Notation (GSN) Format.
I will give you the following information in the form of Questions and Answers:
**Question 1: What is a safety case?**
Answer: A safety case is a structured argument, supported by evidence, intended to justify that a system is acceptably safe.
**Question 2: What is the format of the safety case**
Answer: I want you to generate a safety case in GSN Format
**Question 3: What is the system for which you need to generate a safety case**
Answer: X-ray back-scattering machine
**Question 4: What is the main objective of the safety case**
Answer: Create a safety case which seeks to show that radiation over the standard amount (overradiation) will not occur with an x-ray back-scattering machine, like what one might see at an airport.
**Question 5: What is the evidence available for the X-ray machine**
Answer: Formal verification results and Testing results are available for the software operation and Fault tree analysis and Testing results are available for the Timer interlock operation.
Create a safety case for the X-ray back-scattering machine in GSN format.

#### 4.3.5 Measures used to assess the results

**Measures used to assess the results obtained for RQ3.1** To answer RQ3.1, we conducted two distinct types of assessment. Firstly, to assess GPT-4's responses to each of these 19 questions, we will enlist the expertise of two graduate students, all well-versed in GSN and its associated structured prose. Each researcher will independently score the answers on a linear scale ranging from 1 (completely accurate) to 5 (inaccurate). Subsequently, we will calculate the Kendall's Tau correlation [96], similar to [29], to determine the inter-rater agreement. We utilized an online calculator for this purpose[1]. The value of the Kendall's Tau progresses from -1 to 1. When the value of the Kendall's Tau correlation is close to -1, this means that the level of agreement is almost close to none. However, when the value is close to 1, then the level of agreement is strong. Equation (1) outlines the computation of Kendall's Tau.

$$\tau = \frac{C - D}{C + D} \tag{1}$$

In that equation, $C$ is the number of concordant pairs, while $D$ is the number of discordant pairs.
As part of the second assessment, we sought to determine whether GPT-4 could attain a passing grade concerning its knowledge of the goal structuring notation (GSN). To achieve this, we computed the average scores across the four rounds **(k=4)** to establish the grade achieved by GPT-4.

[1] https://www.gigacalculator.com/calculators/correlation-coefficient-calculator.php

**Measures used to assess the results obtained for RQ3.2** Like [29], in RQ3.2, the assessment is done by two assessors (a graduate student and a faculty member), and meetings were held to resolve any potential conflicts in the assessment process. To answer RQ3.2, we prompted GPT-4 to generate safety cases for each of the 4 experiments. Each experiment consists of **k=4** rounds (runs), and GPT-4 generates one safety case per round. GPT-4 therefore generates a total of sixteen safety cases for each of the three ground-truth safety cases used in the four experiments. We relied on the following measures for the assessment of the safety cases that GPT-4 generated when we performed the 4 experiments:

**Ground-truth similarity:** Using this measure we assess the resemblance between the safety case generated by GPT-4 and the ground-truth safety case. To assess the ground-truth similarity, we rely on the following measures/scores:

1. **Sequence matching score:** The SequenceMatcher class from the difflib library in Python compares two strings and provides a matching score between 0 and 1. Values close to 1 represent high similarity and close to 0 represents dissimilarity [2].

2. **Bleu score:** Bleu score primarily evaluates the similarity of text based on n-grams, which are sequences of words. It doesn't directly measure the semantic similarity between sentences or capture the meaning of the text. Instead, it focuses on measuring how many n-grams in the generated text appear in the reference text. Bleu scores typically range from 0 to 1, where: 0 means no similarity and values close 1 represent high similarity [3].

3. **Semantic similarity:** Here, we rely on the cosine similarity to automatically compute the semantic similarity (resemblance) between the ground-truth of a safety case (in the structured prose) and each safety case generated by GPT-4 and represented in the structured prose. The value of cosine similarity is between -1 and 1 with -1 being no similarity and 1 being similar. To compute cosine similarity automatically, we utilized python scripts.

**Reasonability:** Reasonable means the GSN "*element could reasonably be in the ground-truth but is not*" [29]. To assess the reasonability, we first convert each safety case generated by GPT-4 in the structured prose to GSN diagrams manually. The resulting goal structure is rated by three raters based on the following linear scale: ***1=Totally reasonable; 2=Mostly reasonable; 3=Moderately reasonable; 4=Slightly reasonable; 5=Unreasonable***.

[2] https://www.educative.io/answers/what-is-sequencematcher-in-python

[3] https://cloud.google.com/translate/automl/docs/evaluate

**Python script to compute sequence matching score**

```
from difflib import SequenceMatcher

def similar(a, b):
return SequenceMatcher(None, a, b).ratio()
#ground-truth safety case
with open('GT.txt', 'r', encoding='utf-8') as file:
doc1 = file.read()
#GPT-4 generated safety case
with open('gptresult.txt', 'r', encoding='utf-8') as file:
doc2 = file.read()
sim = similar(doc1, doc2)
print(sim)
```

**Python script to compute bleu score**

```
import nltk
nltk.download('punkt')
from nltk.translate.bleu_score import corpus_bleu

def load_text(file_path):
with open(file_path, 'r', encoding='utf-8') as file:
text = file.read()
return text.strip()
def preprocess(text):
# Tokenize the text
return nltk.word_tokenize(text)
def calculate_bleu_score(reference_text, generated_text):
# Tokenize reference and generated texts
reference_tokens = preprocess(reference_text)
generated_tokens = preprocess(generated_text)
# Calculate BLEU score
return corpus_bleu([[reference_tokens]],[generated_tokens])
def main():
# File paths for the reference and generated safety cases
reference_file = 'GT.txt' ground truth safety case
generated_file = 'gptresult.txt' gpt generated safety case
# Load text from files
reference_text = load_text(reference_file)
generated_text = load_text(generated_file)
Calculate BLEU score
bleu_score = calculate_bleu_score(reference_text, generated_text)
print("BLEU Score:", bleu_score)
if __name__ == "__main__":
main()
```

**Python script to compute cosine similarity**

```
from transformers import BertTokenizer, BertModel
import torch
from sklearn.metrics.pairwise import cosine_similarity

def bert_encode(document, tokenizer, model):
tokens = tokenizer.encode_plus(document, add_special_tokens=True, max_length=512, truncation=True, padding="max_length", return_tensors="pt")
with torch.no_grad():
outputs = model(**tokens)
return outputs[0].mean(dim=1)
def calculate_similarity(doc1, doc2):
# Load pre-trained model tokenizer (vocabulary)
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
# Load pre-trained model (weights)
model = BertModel.from_pretrained('bert-base-uncased', return_dict = False)
# Encode text
encoded_doc1 = bert_encode(doc1, tokenizer, model)
encoded_doc2 = bert_encode(doc2, tokenizer, model)
# Calculate cosine similarity
return cosine_similarity(encoded_doc1, encoded_doc2)[0][0]
with open('GT.txt', 'r', encoding='utf-8') as file:
doc1 = file.read()
with open('gpt_result.txt', 'r', encoding='utf-8') as file:
doc2 = file.read()
similarity_score = calculate_similarity(doc1, doc2)
print("Semantic Similarity Score:", similarity_score)
```

# 5 Results of RQ1: bibliometric analysis

The PRISMA flow diagram that Figure 14 depicts shows the results of each of the six phases that we completed to select the primary studies we included in our bibliometric analysis. That PRISMA flow diagram comprises three main stages (i.e. *Identification*, *Screening*, and *Included*) that map to the five phases of our selection process. Hence, at the *Identification* stage, we started with 6,765 records that we identified from five databases and imported to EndNote. The first stage is the *Identification* stage: it documents the number of records we identified from each source which amounts to 6765. Additionally, this stage highlights the number of records that we eliminated in phase 2, prior to the screening process, such as duplicate records found in other databases, and this resulted in a total of 4709 records. The second stage is the *Screening* stage, which outlines phase 3 and 4 explained in Section 4.1.2. The results of each screening phase are presented here (Phase 3 = 941; Phase 4 = 373), along with the rationale for excluding irrelevant studies. Finally, at the *Included* stage, we selected 224 primary studies that we included in our bibliometric analysis. The list of all 224 primary studies along with the title, author information and venue is present at the Appending A

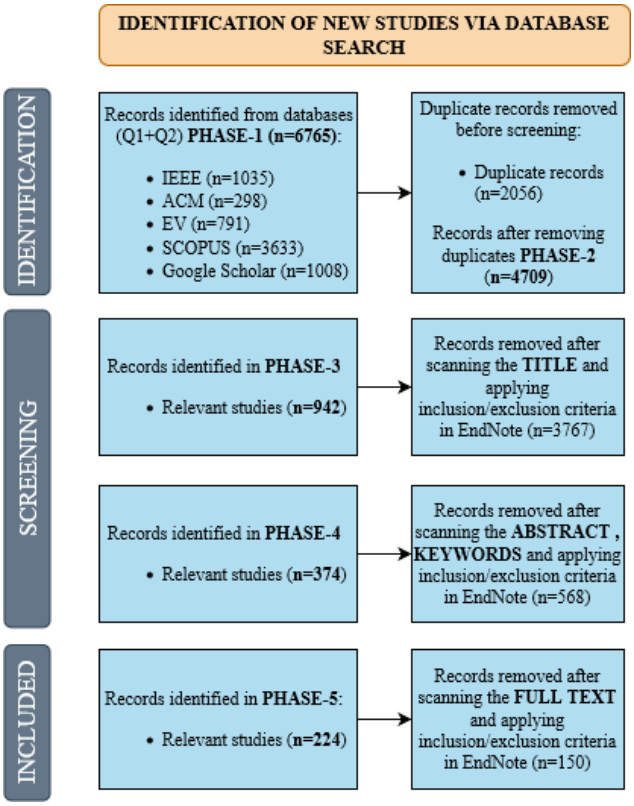


Figure 14: PRISMA Flow diagram of the selection process.

## 5.1 Characteristics of primary studies by year:

Figure 15 illustrates the distribution of primary studies published between 2012 and 2022. This reveals some fluctuations in the number of studies published during that period.

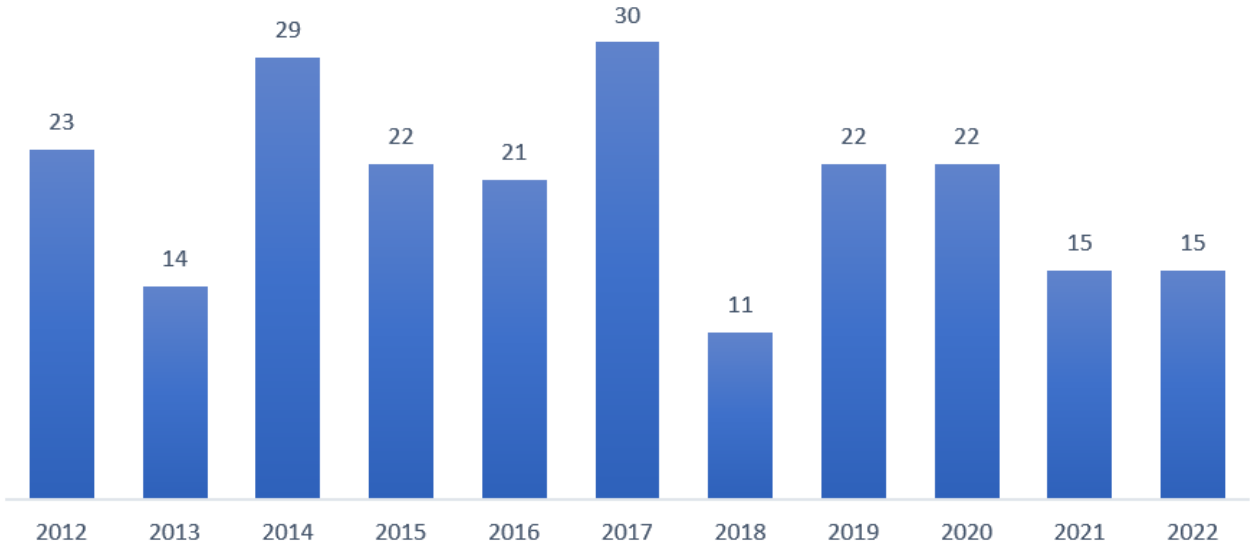


Figure 15: Distribution of primary studies per year (2012-2022)

A notable observation drawn from Figure 15 is the distinct peak in 2017, with a substantial increase in publications. In contrast to the preceding year, which witnessed the publication of only 21 studies, 2017 marked a significant surge with 30 publications. Similarly, there was a surge in publications in 2019 and 2020, with 22 studies being released, compared to 2018 with only 13 studies being published. Despite the decrease in publications during certain years, such as 2013, 2018, and 2021, the subsequent years show an upward trend, indicating a sustained interest in the topic of safety cases. The decrease in 2021 may be due to the COVID-19 pandemic that disrupted research activities across the world and therefore had an adverse impact on the number of publications that year and the subsequent year. Still, with the recent rise of autonomous systems (e.g., autonomous driving vehicles, drones) that need to demonstrate sufficient safe autonomy before being trusted and deployed, we are confident that the research on safety cases will remain critical and attractive and will lead to many publications in the upcoming years.

## 5.2 Characteristics of primary studies by most influential keywords:

We relied on VOSViewer to generate the map that Figure 16 displays. That figure illustrates the most frequent keywords occurring in the 224 studies included in this analysis. To create the map while encompassing a wide range of related keywords, in VOSviewer, we set the minimum occurrence limit for keywords to 2. The analysis of the map reveals several interesting findings. First and foremost, the keywords ***"Safety Case", "Safety Engineering", "Safety Factor", "Safety Arguments", "Safety Case Patterns", "GSN"*** and ***"Safety Goals"*** emerge as the top terms appearing in a significant number of studies. This aligns with the search queries employed during the database-driven search, which specifically targeted literature related to these terms. The prevalence of these keywords indicates that the included studies indeed directly address the topic of safety cases and safety assurance.

The links between the keywords ***"Machine Learning", "Artificial Intelligence", "Autonomous Vehicles",***

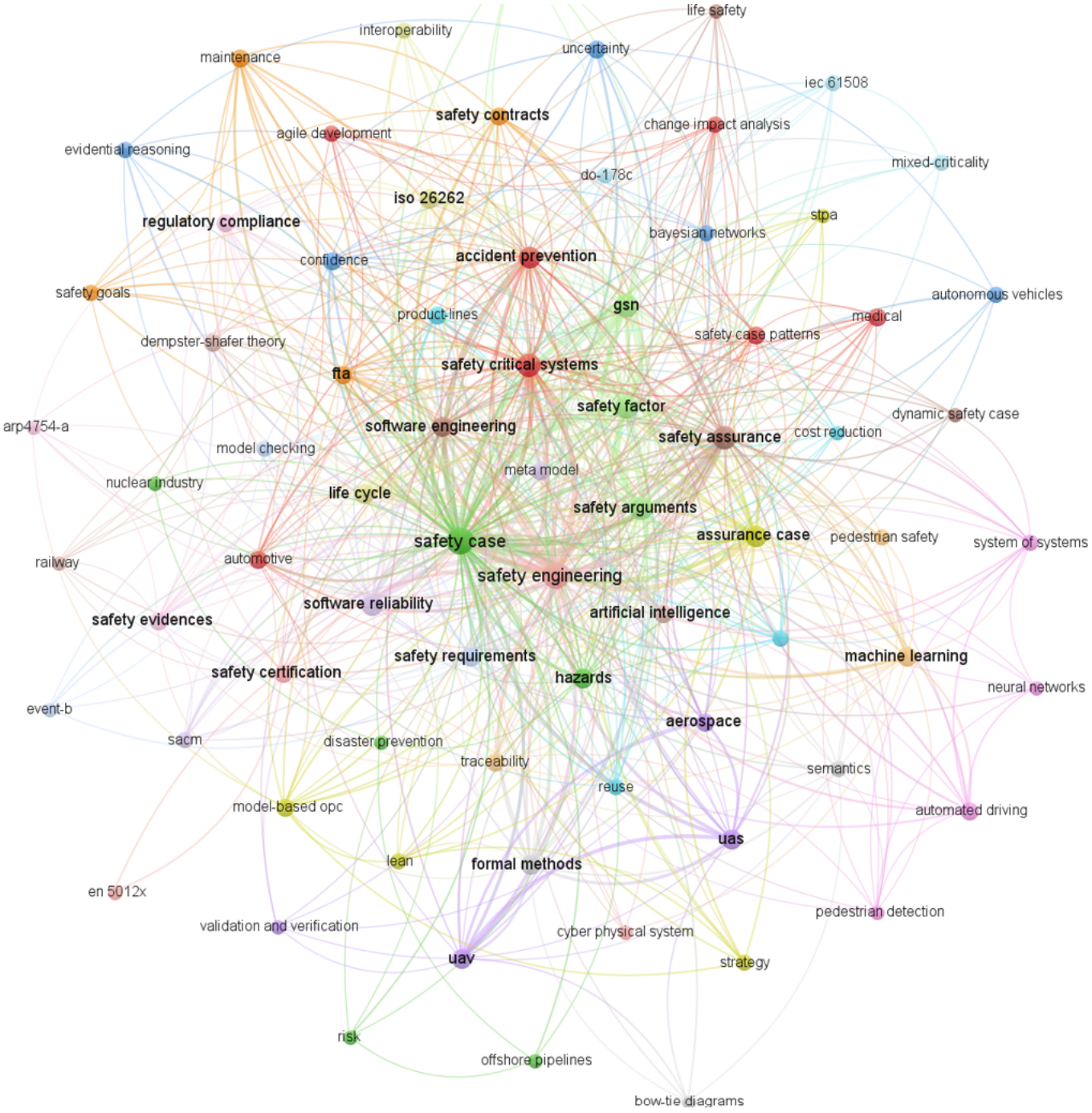


Figure 16: Co-occurence map of most influential keywords.

***"Pedestrian Detection", "Neural Networks"*** and ***"Safety Case"*** highlights the increasing importance of safety assurance for AI-enabled systems (e.g. ML-enabled systems). Over the past few years, especially with the advent of autonomous vehicles, there has been a growing recognition of the need to assure the safety of ML-enabled systems, and this trend is expected to grow further in the future. As ML-based systems gain momentum, assuring their safety becomes crucial. The uncertain and dynamic nature of the environments in which these systems operate introduces potentially risky unforeseen situations. Thus, it is crucial to establish mechanisms to ensure that the critical decisions made by these systems are reliable, adhere to safety standards, and can be thoroughly evaluated for potential risks. Therefore, the demand for safety assurance in ML-enabled systems is of utmost importance to build trust in these systems and mitigate the potential hazards they may face at runtime.

The map highlights the significance of keywords such as ***"Regulatory Compliance" and " Safety Certification"*** in the context of safety assurance. These two concepts are closely intertwined, as certification of safety-critical systems is typically supported by using safety cases to demonstrate compliance with established standards. These safety cases are usually submitted to certification bodies [37]. Co-incidentally, the map also depicts keywords such as ***"IEC 61508", "ARP 4754-A", "ISO 26262", "EN 5012X", "DO-178C"*** which are safety standards used in domains such as Aerospace, Automotive, and Railways. Such standards usually mandate the construction of safety cases to demonstrate a system meets its requirements [37, 35]. Standard compliance and certification are critical in ensuring the safety and reliability of safety-critical systems. Failure to certify such systems may have deadly consequences. This was the case with the Titan submersible, where the company OceanGate failed to adhere to safety standards and obtain proper certification. This eventually led to the deadly implosion of that underwater vessel.

Figure 16 also highlights the importance of approaching safety assurance through the " ***model-driven engineering***" lens. As depicted in the figure, numerous metamodels, languages, and notations have emerged to provide formal frameworks for representing safety cases. Among these, two prominent contenders are the ***"GSN"*** and the ***Structured Assurance Case Metamodel (SACM)***. GSN, which has enjoyed widespread adoption for several years, has historically been the go-to choice for illustrating safety cases. However, in recent times, an increasing number of researchers have transitioned to SACM, an OMG-standardized metamodel for assurance cases [97, 98]. This shift is attributed to SACM's unique qualities, as articulated in Foster et al. [35]: "*SACM not only unifies and extends various predecessor notations, including GSN and CAE but also positions itself as a definitive reference model in the field of safety assurance*".

Traditional safety cases are usually static. This means they are suitable to certify a system before its deployment, but may become incorrect, obsolete or even inadequate at runtime (i.e. during the system operation) [99, 100]. The presence of the keywords ***"Dynamic safety case", "Change Impact Analysis"*** on Figure 15 therefore underscores the importance of continuous and adaptable safety assurance in the face of evolving safety-critical systems. One key factor that necessitates dynamic safety assurance is the presence of ***"Uncertainty"*** to which safety-critical systems have to face during their operations. Numerous assessment techniques have emerged as valuable tools for effectively evaluating and quantifying uncertainty, as well as confidence levels, within safety cases. Three notable approaches are Bayesian analysis utilizing ***"Bayesian Networks", "Evidential Reasoning", and the "Dempster-Schafer theory"***, all of which are highlighted in

Figure 15. Safety-critical systems operate in complex and unpredictable environments, where new risks and challenges can emerge over time, making the dynamic safety assurance critical. Uncertainty can arise from various sources, such as changes in system requirements, advancements in technology, or unexpected interactions with the environment or users. To effectively address that uncertainty, safety assurance must be an ongoing process rather than a one-time activity. Dynamic safety cases allow for continuous monitoring, assessment, and adaptation of safety measures to ensure that safety requirements are met in the evolving context.

Traditional safety cases are static i.e., only suitable for certifying systems before they are deployed. safety cases may become obsolete during system operation [99, 100]. The presence of keywords such as ***"Dynamic Safety Case"*** and ***"Change Impact Analysis"*** in Figure 16 highlights the need for an adaptable safety assurance process allowing to monitor, assess, and adapt safety measures as a system evolves or its operational contexts change. The need for dynamic assurance is further compounded by the ***"Uncertainty"*** safety-critical systems face at runtime. That uncertainty is usually quantified by relying on assessment techniques like ***"Bayesian Networks"***, ***"Evidential Reasoning"***, and ***"Dempster-Shafer Theory"***.

The keywords ***"Automotive", "Autonomous Vehicles"*** both indicate that safety assurance plays a crucial role in the automotive domain, particularly in the context of autonomous vehicles. Such vehicles hold the promise of revolutionizing transportation by offering increased convenience, improved efficiency, and enhanced safety. So, assuring the safety of these vehicles is of paramount importance, as they operate without direct human intervention and rely on complex algorithms and sensors to make critical decisions on the road. The need to assure such vehicles has translated in a surge of primary studies. Their line of research aligns with the business needs of several companies (e.g., Tesla, Ford, Waymo, and General Motors) that are racing to become leaders in the production of autonomous driving technologies. For instance, Tesla's Robotaxi service [101] and General Motors' Cruise [102] emphasize safety in autonomous vehicle development, reflecting the importance of safety assurance in the development of their products.

For example, Tesla's emphasis on safety assurance is indeed a key component of their marketing strategy. Tesla's Autopilot system, which offers semi-autonomous driving capabilities, is built upon advanced sensor technology, machine learning algorithms, and extensive data collection. By promoting their vehicles with the tagline "Safety first design," Tesla aims to instill confidence in potential customers regarding the safety of their autonomous driving features. Tesla's announcement of the Tesla Robotaxi service [101], further underscores the need for comprehensive safety measures to ensure the safe operation of autonomous vehicles as part of a ride-hailing network. General Motors' Cruise [102] is another notable example, focusing on the development of self-driving cars. Cruise has been working on refining their autonomous driving technology and has made significant strides in testing their vehicles in real-world scenarios. The competition among these automotive companies to achieve technological advancements in autonomous vehicles while maintaining safety standards highlights the criticality of safety assurance. The race to develop autonomous vehicles is not solely about achieving technological milestones but also about building trust with regulators, the public, and potential customers. That race has translated in a surge of primary studies focusing on the safety assurance of autonomous vehicles.

Keywords such as ***"Aerospace", "UAS"*** and ***"UAV"*** indicate the importance of safety assurance in the field

of aerospace. In research, ensuring the safety of Unmanned Aerial Vehicles (UAVs) has been a significant focus [103, 104, 105, 106]. This translated in a quite high number of studies in that domain. The growing use of UAVs in applications like surveillance, delivery, and agriculture highlights the need for robust safety assurance. That line of research aligns with the business needs of companies like Amazon, that needs to assure the safety of its devices (e.g., Prime Air delivery drones [107]).

The increasing use of UAVs in various applications, such as aerial surveillance, package delivery, and agricultural monitoring, necessitates rigorous safety assurance measures. In the industry, companies like Amazon are revolutionizing product delivery through the development of delivery drones under their Amazon Prime Air program [107]. These drones aim to deliver products to customers within 30 minutes of their online purchase. However, the success and widespread adoption of such drone delivery systems rely heavily on ensuring their safety. Aerospace safety assurance is of paramount importance in both research and industry, particularly in the context of UAVs and delivery drones. The need to assure the safety of these aerial systems is crucial for their successful integration into everyday operations. This explains why several primary studies focus on that topic.

On the map (Figure 16), keywords like ***"Medical", "Railway"***, ***"Nuclear"***, and ***"Offshore pipeline"*** indicate that the safety case trend is also gaining momentum in health (e.g., applications in infusion pumps) , railway (e.g., autonomous trains), maritime (e.g., autonomous ships) and nuclear domains (e.g., nuclear power plants).

Overall, this map visually illustrates the relationships and connections between the most influential keywords. It provides insights into the associations and co-occurrences among the identified terms. The visualization aids in identifying the key focus areas within the research domain and offers a foundation for further analysis and interpretation of the underlying literature.

### 5.3 Temporal evolution of most influential keywords:

The Sankey diagram that Figure 17 depicts, shows the temporal evolution of the most influential keywords depicted in Figure 16. We generated this diagram by using Google Charts [89]. Each block within the Sankey diagram represents the most frequently occurring keyword and its associated sub-period.[4].

The Sankey diagram in Figure 17 offers insights into the evolution of safety cases. The application domains of safety cases have evolved, with initial use (2010s) in aerospace and medical fields expanding to include automotive, railway, and, more recently, the oil engineering sector, with a significant focus on autonomous vehicle technologies. In the early 2010s, static safety cases were developed using ***formal methods (2010s)***. At some point, the focus of these methods has shifted mainly to notations such as ***GSN (2016s)***. Over the time, other categories of safety cases have emerged. First, ***Modular safety cases*** emphasizing modularity. Then, in the years 2020-2022, ***Dynamic safety cases*** and ***Agile safety cases*** have emerged to cope with the complexity and heterogeneity of safety-critical systems (e.g., CPSs). Agile safety cases adopt ***Agile development*** and ***Lean*** practices.

The Sankey diagram also sheds light on the temporal evolution of safety assurance, in the context of standards

[4]The connections between blocks and their respective thicknesses allow quantifying the interrelationships between different keywords across various time periods. A thicker connecting line indicates a stronger association between the two concepts [19]

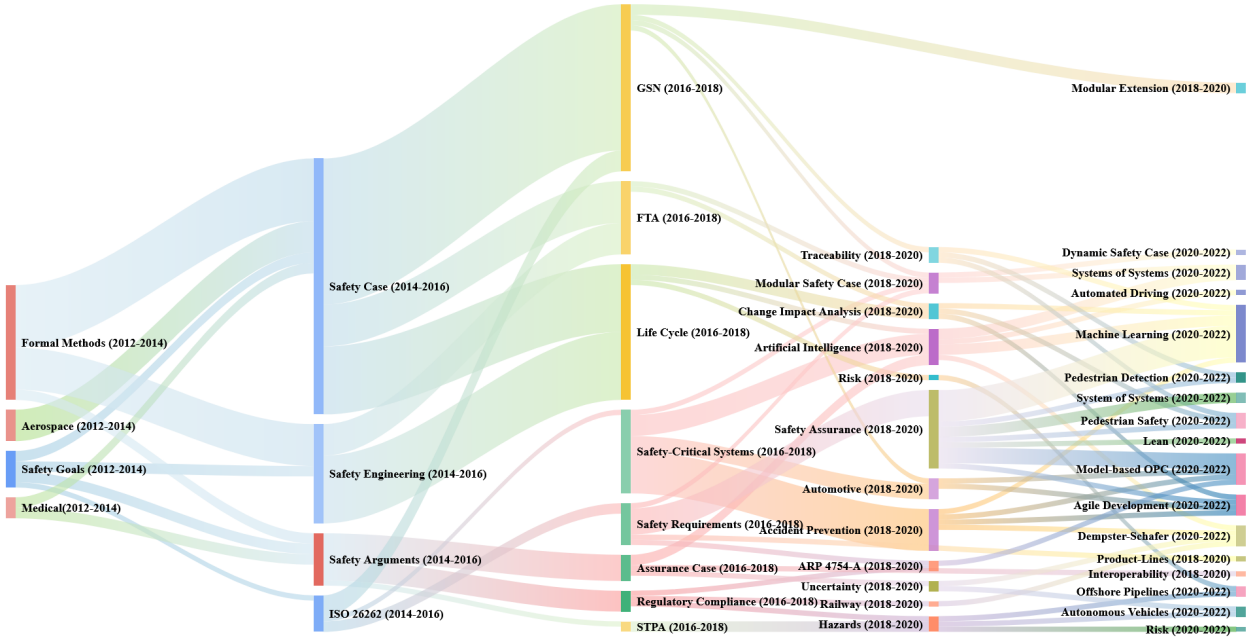


Figure 17: Temporal evolution of the most influential keywords from 2012-2022.

and their adoption in specific domains. Around the year 2014, there was a notable surge in the attention given to ***ISO 26262***. This standard, primarily focused on functional safety in road vehicles, gained prominence as safety cases found their way into the automotive industry. ISO 26262 played a pivotal role in setting safety benchmarks and guidelines for this domain. In the subsequent years, the emphasis on regulatory compliance became increasingly pronounced. This trend indicates a growing recognition of the importance of adhering to industry-specific regulations and standards to ensure safety. Compliance with these standards is crucial, particularly in safety-critical domains such as automotive and aerospace. The keyword "***automotive***" gained momentum in the years following regulatory compliance, indicating a continued focus on safety within the automotive sector. This likely reflects the ongoing integration of safety measures, including the use of safety cases, during the development and operation of vehicles. Additionally, the emergence of a new standard, ***ARP 4754-A***, specifically designed for civil aviation, underscores the need for domain-specific safety standards. This standard likely addresses the unique challenges and safety requirements within the aerospace domain, further emphasizing the importance of tailored safety measures in different industries.
Figure 17 also illuminates the emergence of AI and ML within the safety assurance landscape, primarily driven by the uncertainty inherent to the unpredictability of the operational contexts in which CPSs operate. Notably, all three keywords—***AI, ML, and CPSs***—are present within the 2020-2022 timeframe, signifying not only a current upward trajectory but also a strong indication of their anticipated prominence in the future research.

## 5.4 Characteristics of primary studies by venues

The bar chart in Figure 18 depicts the top 10 venues of the primary studies included in our bibliometric analysis. The SAFECOMP Conference (International Conference on Computer Safety, Reliability, and Security)

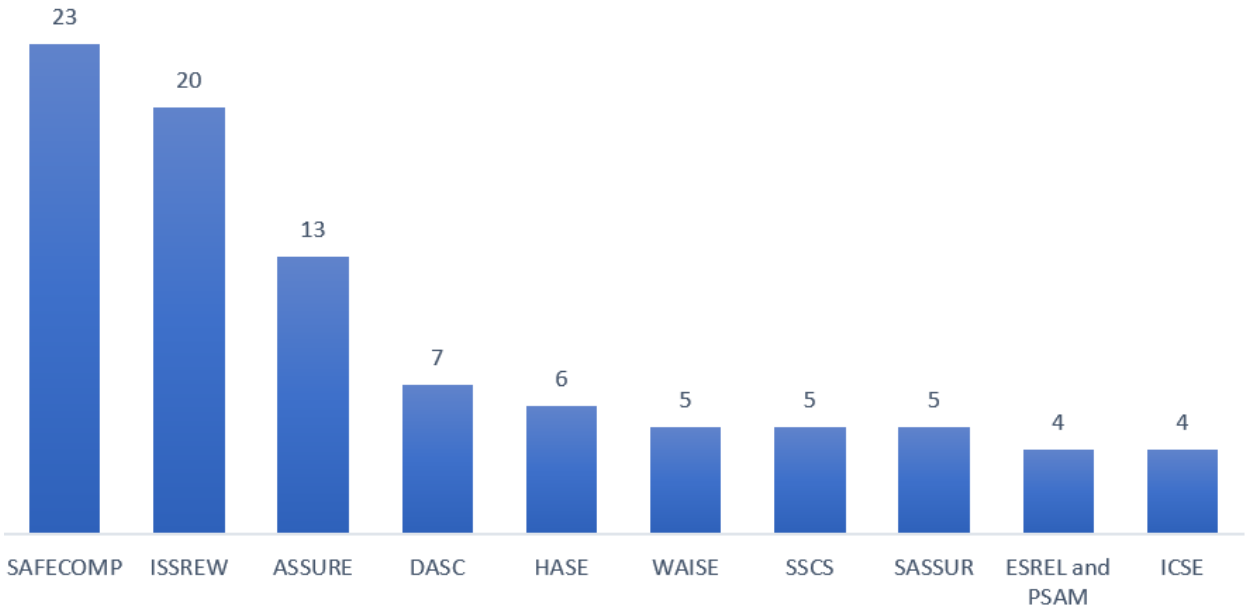


Figure 18: Bar chart showing the top 10 primary venues.

emerges as the leading venue, featuring 23 studies. The rationale is that SAFECOMP has an exclusive dedication to safety assurance. By focusing solely on that aspect, SAFECOMP provides a specialized platform for researchers, practitioners, and experts to share their insights and advancements in the field of safety assurance. Following closely is ISSREW (International Symposium on Software Reliability Engineering Workshop), with 20 studies, and ASSURE (International Workshop on Assurance Cases for Software-Intensive Systems) securing the third position with 13 publications. This trend indicates that a significant number of studies on safety cases are published in conferences and workshops that focus on safety assurance. It is worth noting that workshops such as ASSURE, WAISE (International Workshop on Artificial Intelligence Safety Engineering), and SASSUR (International Workshop on Next Generation of System Assurance Approaches for Critical Systems) are conducted as integral components of the SAFECOMP conference. This observation reinforces the significance of SAFECOMP as a key conference for the dissemination of research on safety assurance. Abbreviations and Acronyms of the all the conferences and journals are listed in Appendix B.

## 5.5 Characteristics of primary studies by citations in Google Scholar

Similar to Deng et al. [23], we also conducted a bibliometric analysis to identify the most cited primary studies. Table 5[5] presents the 15 most cited studies based on their number of citations.

Two of these studies (i.e. [113] and [114]) jointly scored the highest number of citations (i.e. 107). Denney et al. are the authors of [113]. That study introduces AdvoCATE, a tool designed to automate the creation and management of safety assurance arguments, with a specific focus on unmanned aircraft systems (UAS). Birch et al. are the authors of [114]. That study explores the role of safety cases in the functional assessment

[5]We created Table 5, based on 219 studies, as five of the studies (i.e., [108], [109], [110], [111], [112]) were not accessible on Google Scholar at the time we searched them.

Table 5: Top 15 most cited primary studies from Google Scholar as of September 23, 2023.

| No | Author | Title | # Citations |
|---|---|---|---|
| 1 | Denney et al. (b27) | Tool support for assurance case development | 107 |
| 1 | Birch et al. (b21) | Safety cases and their role in ISO 26262 functional safety assessment | 107 |
| 3 | Denney et al. (b5) | AdvoCATE: An assurance case automation toolset | 104 |
| 4 | Denney et al. (b28) | Dynamic Safety Cases for Through-Life Safety Assurance | 98 |
| 5 | Denney et al. (b29) | A formal basis for safety case patterns | 68 |
| 6 | Yamamoto et al. (b30) | An evaluation of argument patterns to reduce pitfalls of applying assurance case | 61 |
| 7 | Gallina (b31) | A model-driven safety certification method for process compliance | 59 |
| 8 | Nair et al. (b32) | An evidential reasoning approach for assessing confidence in safety evidence | 55 |
| 9 | Dardar et al. (b33) | Industrial experiences of building a safety case in compliance with ISO 26262 | 51 |
| 10 | Denney et al. (b34) | A lightweight methodology for safety case assembly | 48 |
| 11 | Denney et al. (b35) | Automating the assembly of aviation safety cases | 48 |
| 12 | Ayoub et al. (b36) | A systematic approach to justifying sufficient confidence in software safety arguments | 46 |
| 13 | Weinstock et al. (b37) | Measuring assurance case confidence using Baconian probabilities | 45 |
| 14 | Ayoub et al. (b38) | A safety case pattern for model-based development approach | 44 |
| 15 | Guiochet et al. (b39) | A model for safety case confidence assessment | 43 |

process defined by the ISO 26262 standard.

Interestingly, the studies that Table 5 reports mostly focus on critical topics such as the development of safety cases, their assessment, and the use of patterns to foster the reuse of safety case argument structures. They also focus on dynamic safety assurance, the use of safety cases in compliance with industrial standards (e.g., ISO 26262), and the development of tools automating the safety assurance process.

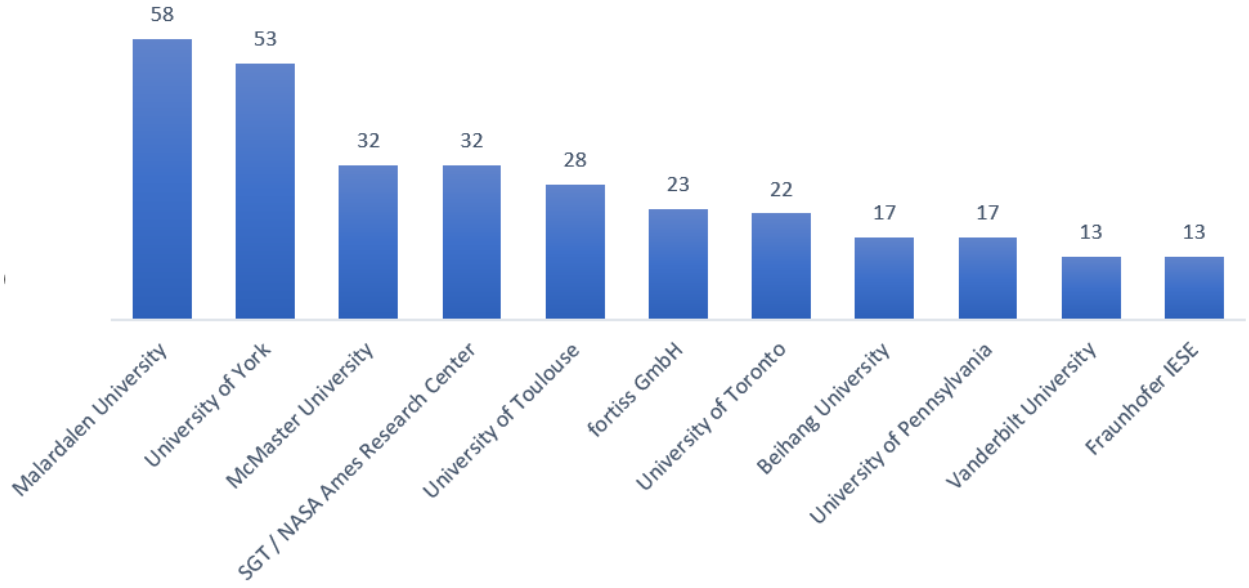


Figure 19: Bar chart showing the top affiliations of authors.

## 5.6 Characteristics of primary studies by author's affiliation:

Figure 19 displays the top organizations and universities based on the number of authors contributing to the field of safety assurance, among the 224 primary studies included in this study. Notably, Malardalen University from Sweden stands out as the leading institution with 58 authors affiliated with that university. The University of York (United Kingdom) follows closely in second place with 53 authors, while McMaster University (Canada) and SGT/NASA Ames Research Center (USA) share the third position with 32 researchers affiliated with these organizations. The University of Toulouse arrives at the fifth position of that ranking with 28 authors.

It is interesting to analyze the findings of this chart in relation to what Figure 20 depicts. In Figure 20, the USA ranked first with authors contributing 155 times, indicating a significant overall contribution from various American organizations. However, when considering the specific organizations, we observe that no single affiliation from the USA secures the top position in Figure 19. This is due to the heterogeneous nature of contributions from multiple organizations in that country. On the other hand, Malardalen University's first-place position in Figure 19 can be attributed to the concentrated efforts of the university itself, as it emerges as a prominent contributor in the field of safety assurance. Such insights shed light on the diverse and multifaceted nature of safety assurance research and the varying roles played by different organizations

in advancing the field.
Interestingly, thanks to Figure 19, researchers can pinpoint organizations that have shown a significant commitment and expertise in safety assurance, and identify potential collaboration opportunities with experts affiliated with these institutions.

## 5.7 Characteristics of primary studies by author's country of affiliation:

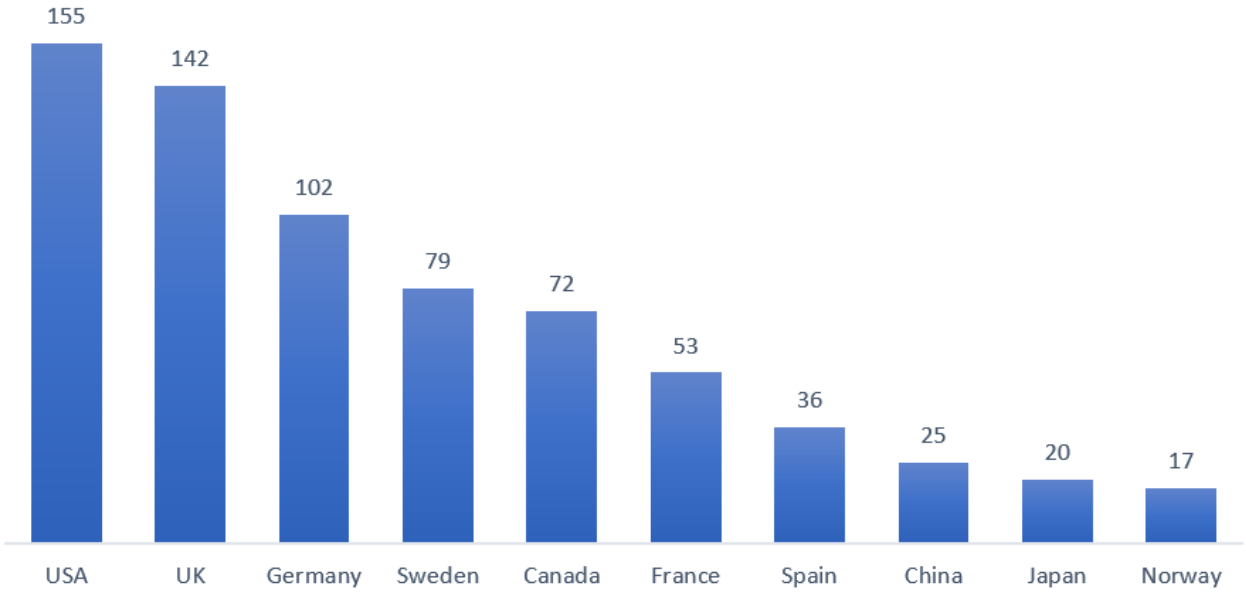


Figure 20: Bar chart showing the top countries of affiliations.

Figure 20 illustrates the top 10 countries of author's affiliations across the 224 studies included in our analysis. The United States (USA) emerges as the top country, with 155 authors who have contributed to safety case literature during the past decade. The United Kingdom (UK) arrives at the second position, with 142 authors. Germany arrives at the third position, with 102 authors.
The prominence of the USA, UK, and Germany may be attributed to the fact that these countries have established numerous standards such as ISO 26262 [14], DO-178C [115], and EUROCAE [116], which mandate the development of safety cases for a given system. Consequently, authors from these countries contribute extensively to the topic due to the prevalence of safety standards and processes in their respective countries. Furthermore, the high occurrence of safety-related disasters (e.g., Columbia shuttle disaster, accidents caused by autonomous driving cars, the recent Titan submersible disaster), in some of these countries may also contribute to the increased focus on developing state-of-the-art techniques on safety cases. These countries recognize the critical need for advancing safety practices and, as a result, attract significant contributions from authors in the field. With the volume of contributions depicted in Figure 20, the top leading nations could become the go-to nations for other nations who may be interested in formulating safety case policies.
Figure 20 reveals notable differences in the contribution of authors from various countries to the field of

safety cases. For instance, countries such as Japan and Norway are shown to have relatively fewer authors who have actively contributed to this domain. Several interpretations can be made regarding this observation. For example, in Japan, safety disasters such as the Fukushima Daiichi nuclear accident may have underscored the critical need for robust national and international safety standards and guidelines. While the number of studies with respect to safety cases might not be as high as its other counterparts such as Sweden, France, China, etc., the country has implemented alternative safety measures and standards to prevent and address such safety disasters in the future [117].

Similarly, in the case of Norway, while safety cases may not be widely practiced, the country has a strong focus on safety measures. Norway has stringent regulations and safety standards to ensure the protection of its natural environment and the safety of workers in these sectors [118]. The emphasis on safety and the presence of well-established safety protocols demonstrate a different approach to safety assurance compared to safety cases.

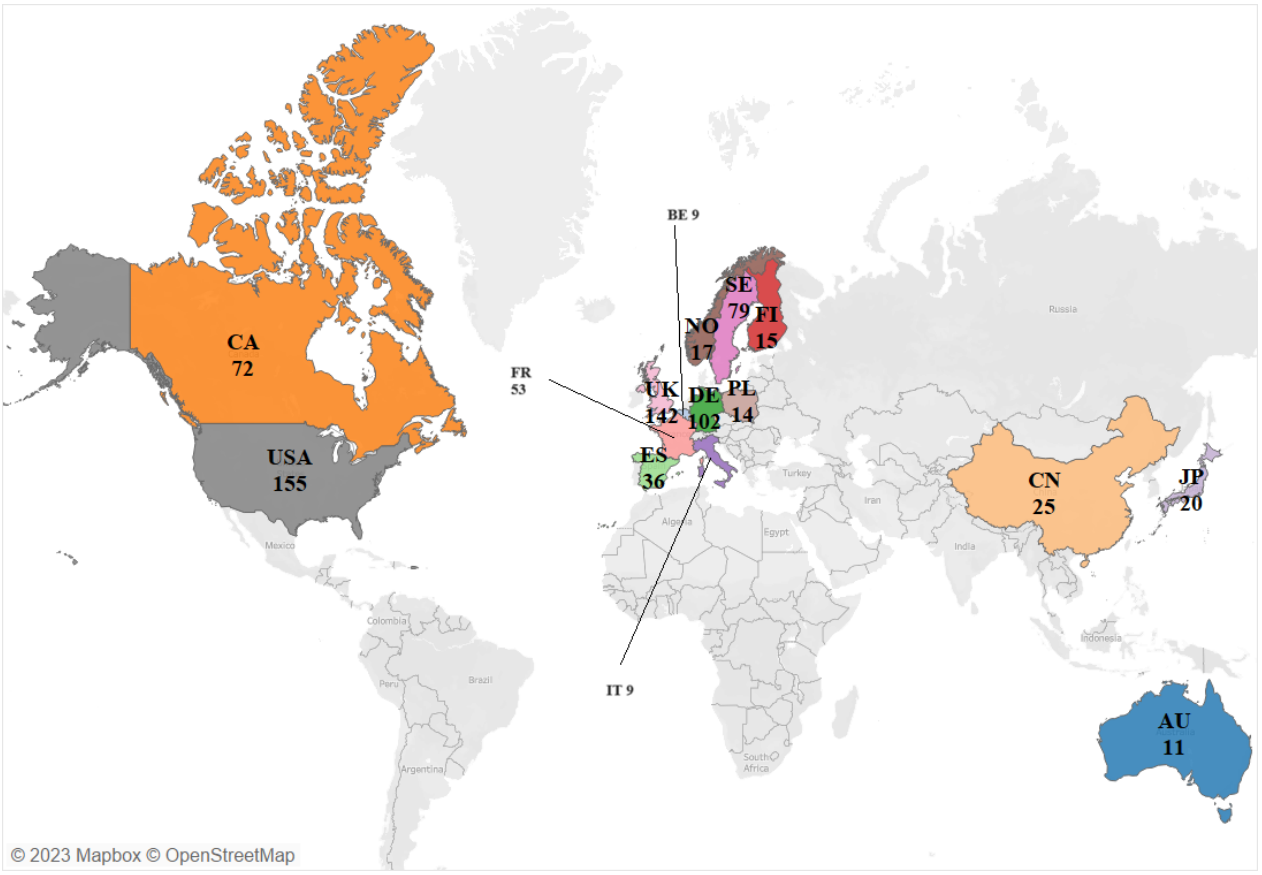


Figure 21: World Map showing the top 15 countries.

To complement the bar chart (Figure 20), we used Tableau [90] to create a world map that shows the top 15 countries with the highest number of contributions from authors in Figure 21 [6].

Figures 20 and 21 both illustrate a pattern of moderate diversity within the landscape of safety case literature.

[6]In that Figure, USA = United States of America; UK = United Kingdom; DE = Germany; SE= Sweden; CA = Canada; FR= France; ES= Spain; CN= China; JP = Japan; NO = Norway; FI = Finland; PL= Poland; AU= Australia; IT = Italy; BE= Belgium

More specifically, the majority of the literature originates from two continents, namely North America and Europe, with their respective institutions significantly represented in this domain. North America, particularly the USA and Canada, emerges as a focal point for safety case research. Interestingly, Figures 20 and 21, also underscore China's growing influence and contribution to the field of safety assurance in Asia.
Still, some countries from Asia, Oceania, South America and Africa are not explicitly depicted in Figures 20 and 21. While some of these countries play significant roles in the global economy, their absence may be attributed to several factors. First, authors with nationalities of these countries and who are actively involved in safety case research may be affiliated with institutions or organizations in foreign countries, such as the USA, UK, or Canada. Therefore, their contributions might be listed under these foreign countries. Second, safety assurance practices can vary based on regional priorities and industries. Addressing this moderate diversity in safety cases literature requires fostering global collaboration, and encouraging research initiatives in under-represented regions.

# 6 Results of RQ2 - Manual creation of a safety case for the system presented in Yuan et al. [1]

To manually create the safety for the for system presented in [1], we meticulously followed the phases in our framework explained in Section 4.2. In the following sub-sections, we explain how we performed the activities mentioned in each of phases in Figure 3 and also document the results of these phases. Note that we only perform first two phases of our framework and document the complete safety case here. The third phase only applies whenever there is a change in the system of Yuan et al. [1].

## 6.1 Overview of the system

QCar, the flagship vehicle of the Self-Driving Car Studio, embodies an open-architecture design, serving as a scaled model vehicle tailored for academic instruction and research endeavors. Outfitted with a comprehensive array of sensors such as LIDAR, 360-degree vision, depth sensor, IMU, encoders, and expandable IO capabilities, the vehicle offers versatile sensing capabilities. Propelled by an NVIDIA® Jetson™ TX2 supercomputer, QCar delivers remarkable speed and power efficiency, empowering users with high-performance computing capabilities [91]. Figure 22, taken from [91] shows the image of a Qcar with all its features. Yuan et al. [1] implemented the deep reinforcement learning based algorithm, specifically, D3QN PER on the Quanser Qcar for collision avoidance in an unsignalized 4-way intersection scenario.

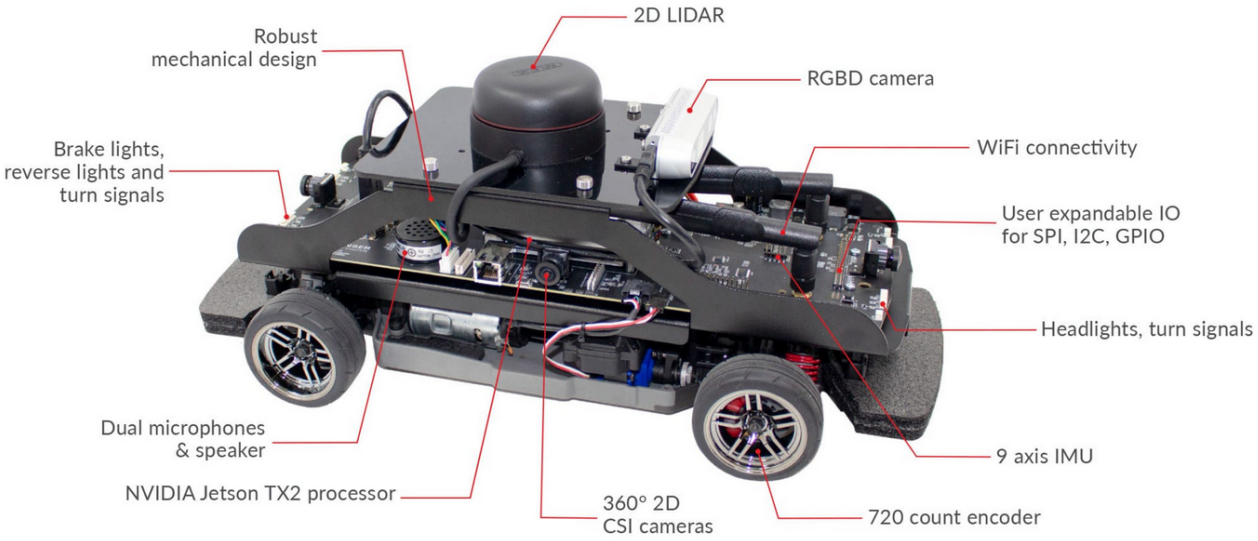


Figure 22: Quanser Qcar with its features explained

## 6.2 Phase 1 - HARA

We performed the hazard analysis and risk assessment (HARA) by following the guidelines presented in ISO 26262-3:2018 [5]. The results of HARA are documented in an Excel template and it is available on

GitHub [7]. At the end of HARA, we have identified the following two safety goals (denoted as HARA-SG) as follows:

1. **HARA-SG1**: Collision of self-driving vehicle with other vehicles shall be prevented
2. **HARA-SG2**: Unintended acceleration of self-driving vehicle shall be prevented

We will utilize these safety goals as input to the AMLAS **stage 1 - ML assurance scoping** and begin to construct the actual safety case in the following sections.

## 6.3 Phase 2 - Apply AMLAS stages

In this phase, we applied the AMLAS stages 1,2, and 6 as explained in Figure 4.2 and obtained the corresponding artefacts and safety arguments. In total, we present 10 artefacts and 3 safety arguments which constitute the safety case for the ML component presented [1].

### 6.3.1 Stage 1 - ML assurance scoping

The main objective of this stage is to determine the scope of the safety assurance process and scope of the safety case for the ML component. Additionally, to create the top-level safety argument specifying the contextual information.

**Artefact [A] – System safety requirements:** We generated the system safety requirements from the safety goals derived at the end of the hazard analysis and risk assessment process. These requirements are listed below:

1. **SYS-SAFE-REQ1** : The self-driving vehicle will start to decelerate if a collision with other car is detected based on the observation.
2. **SYS-SAFE-REQ2** : The self-driving vehicle will not cross the intersection if it observes that other car is in the collision area

The purpose of the proposed D3QNPER RL algorithm [1] is to make sure that collision is avoided when the self-driving vehicles are in the collision area. So, when a car enters the collision area, other cars choose the decelerate action based on their observation.

**Artefact [B] – Description of the System Environment:** In order to determine the safety requirements that are allocated to the ML component, according to AMLAS, it is important that we describe the system environment. The operational design domain (ODD) of the proposed D3QNPER RL algorithm, as explained in Yuan et al. [1], is an unsignalized 4-way intersection. The reason they chose this scenario because this is complex than other traffic scenarios where each car chooses to enter the intersection area and the drivers constantly observe and interact with other road users in order to cross the intersection safely. This experiment

[7] https://github.com/msivakum16/Safety_Case/blob/main/Hazard_analysis_risk_assessment(1).xlsx

was carried out in an indoor, controlled simulation area by the use of Quanser Qcars [91]. There are no other objects present in this scenario.

There are 12 possible paths and maximum of 4 cars passing through the intersection at a given time. Figure 23, taken from [1] represents this scenario. Among the 4 cars, car 1 is the ego vehicle and rest of the cars are considered opponent vehicles. The ego vehicle is trained on the D3QN PER RL algorithm so that it can take action based on the opponent's policy levels. The ego vehicle can go left, right or straight whereas all the opponent vehicles go straight, and they have two optional training policies. These policies are conservative (level-1) and aggressive (level-2). It is to be noted that these behaviours are unknown to the ego vehicle. The opponents can also choose randomly from five pre-defined velocities (0.3 m/s,0.6m/s, 0.9 m/s,1.2m/s, and 1.5m/s) [1].

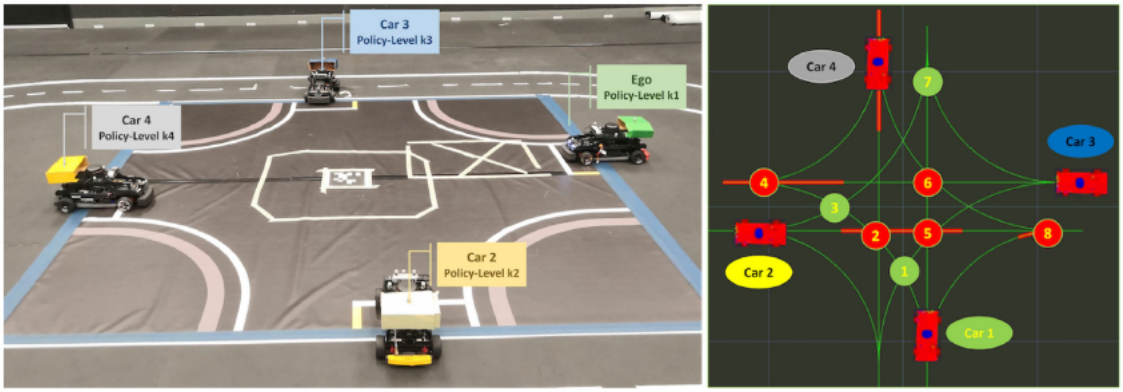


Figure 23: Figure taken from [1] represents the four way intersection: Car 1 is ego car; Car 2, 3 and 4 are opponents with different reasoning levels. Number 1 to 8 are the collision areas from the perspective of ego.

**Artefact [C] – System Description** Figure 24 is adapted from Yuan et al. [1]. That figure provides the end-to-end control scheme. As explained in Yuan et al. [1], the process begins by extracting features from 2D Lidar point cloud data and car states using Long Short-Term Memory (LSTM) to generate observation information. Subsequently, the Q value is produced through a fully connected layer with a dueling structure. An enhanced algorithm, named D3QN PER, is introduced by amalgamating the strengths of traditional Deep Q Network (DQN), Double DQN, and Prioritized Experience Replay (PER) algorithm. Finally, the optimal action in each state is determined by selecting the maximum Q value. Given that the intersection can be represented by 12 paths, steering commands are generated using the pure pursuit controller.

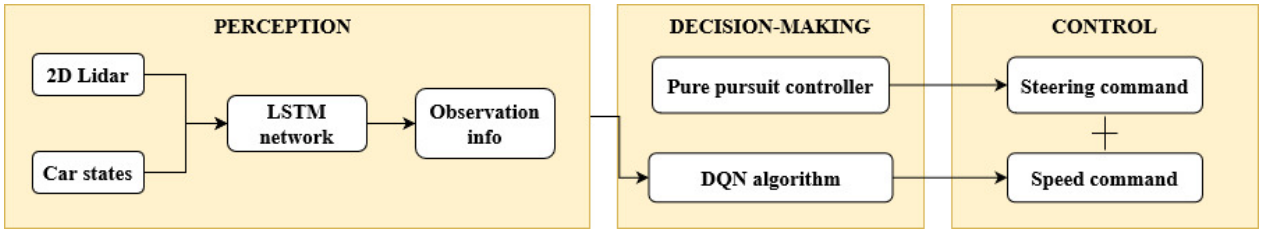


Figure 24: End to end control scheme block diagram adapted from [1]

It is to be noted that Yuan et al. implemented their algorithm on a Quanser Qcar [91] as explained in Section 6.3.1.

**ARTEFACT [D] – ML Component Description** The driver interaction model which Yuan et al. [1] developed, enables the modeling of driver-to-driver and driver-to-autonomous vehicle interactions by combining the use of level-k reasoning with Dueling Double Deep Q Network with Prioritized Experience Replay Algorithm (D3QN PER) algorithm [1].

**Level-k reasoning:** To mimic the decision-making of humans, Yuan et al. [1] utilized a game theoretical concept named level-k reasoning. The basic gist of level-k reasoning is that it presumes that different human beings have different reasoning levels. Here, Yuan et al. [1] explains that level-0 is the lowest level of reasoning and level-3 is the highest level. The following are the descriptions of different levels of reasoning.

1. **Level – 0**: These are called naive agents. Their actions do not depend on other's actions rather it is based on a set of pre-determined moves.
2. **Level – 1**: These agents assume that all other agents have level-0 reasoning and determines its actions based on this assumption.
3. **Level – 2**: Similarly, these agents assume all other agents are level – 1
4. **Level – 3**: These agents assume all other agents are level – 2
5. **Adaptive policy** – This policy combines the advantages and of level-1 and level-2 policies.

This process continues for subsequent levels. Hence, to generalize, we can say that the action of level k agent depends on level k-1 except level 0.

**Dueling Double Deep Q Network with Prioritized Experience Replay Algorithm (D3QN PER)**: Yuan et al. [1] chose to use a combination of three algorithms namely Dueling DQN, Double DQN and PER called as D3QN PER RL algorithm in order to train their ego vehicle. This combination is proved to be more efficient than the three algorithms used alone. The deep Q network (DQN) algorithm, employs Q-learning to generate labeled samples for the deep Q-network. Nonetheless, both Q-learning and DQN suffer from an overestimation issue, as the max operator utilizes identical values for both action selection and evaluation. Double DQN addresses this concern. Prioritized experience replay (PER) operates on the premise that certain experiences hold greater significance for training than others. Rather than random selection, experiences with substantial errors between the deep Q-network and target network are prioritized [1].

**Combination of level-k with D3QN PER algorithm**: To generate vehicles with different levels of reasoning, Yuan et al. [1] ran the D3QN PER RL algorithm in the ROS-Gazebo simulator, where the ego vehicle is the level-k learner. Based on the level-k theory explained above, they assigned the trained level-(k-1) policies (or predefined level-0 behavior) to the all the other vehicles.

**ARTEFACT [E] : Define the Safety Assurance Scope for the ML Component** From the safety goals we identified in phase 1 (HARA), in this section we define the safety requirements that are allocated to the ML component. Based on the HARA, we identified two safety goals:

1. **HARA-SG1**: Collision of self-driving vehicle with other vehicles shall be prevented
2. **HARA-SG2**: Unintended acceleration of self-driving vehicle shall be prevented

Both of these safety goals are geared towards avoiding collision with other cars. Hence, we conclude that the ML safety requirements will mitigate hazard collision (i.e., collision with other vehicles leading to life-threatening injuries).

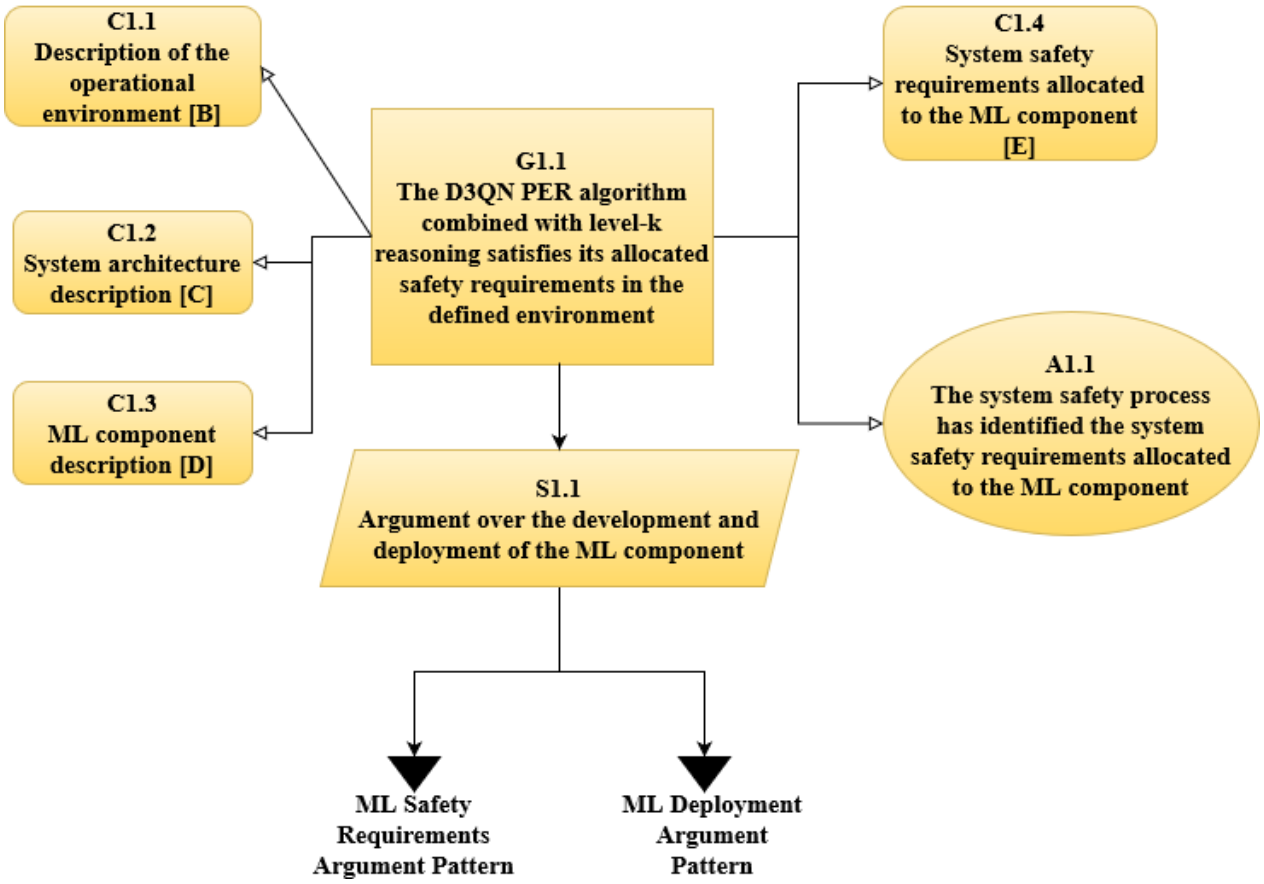


Figure 25: ML safety assurance scoping argument [G] adapted from AMLAS [3] and modified according to our system.

**ARTEFACT [F] : ML safety assurance scoping argument pattern:** Figure 25 explains the top level safety argument. The main objective of this argument is to demonstrate that the D3QN PER algorithm when combined with level-k reasoning is able to satisfy the safety requirements in the defined operating environment which is represented by **G1.1**. This goal is supported by four different contexts C1.1,C1.2,C1.3,C1.4, all of which link to the artefacts [B], [C], [D], and [E] respectively which we have defined above. We make an explicit assumption represented by **A1.1** that we have identified the safety requirements allocated to the

ML component correctly based on the HARA process from phase 1. Finally, we support **G1.1** by an argument **S1.1** which is then split into two parts: ML safety requirements argument pattern and ML deployment argument pattern which will be presented in the subsequent stages.

### 6.3.2 Stage 2 – ML Safety Requirements Assurance

The main objective of this stage is to derive the machine learning safety requirements based on the system safety requirements and validate them. Additionally, to create a safety argument based on the safety evidence generated by the above two objectives.

**Artefact [H] : Develop ML safety requirements** In order to determine this artefact, we require the artefact [E] (safety requirements allocated to the ML component) from the previous stage. Based on the system safety requirements described in artefact [A], and the output of HARA from phase 1, we derived the following ML safety requirements:

1. **ML-REQ1**: The self-driving vehicle trained on the D3QNPER combined with level-k reasoning will decelerate if it detects other car in the collision area first, based on the observation
2. **ML-REQ2**: The self-driving vehicle trained on the D3QNPER combined with level-k reasoning will not pass the intersection first if it detects other car in the collision area.
3. **ML-REQ3**: The self-driving vehicle trained on the D3QNPER combined with level-k reasoning will not attempt to cross the intersection first if it adapted a lower-level policy compared to other vehicles.

We derived ML-REQ1 from the SYS-SAFE-REQ1 and ML-REQ2 and ML-REQ3 from SYS-SAFE-REQ2. ML-REQ1 deals with the ability of the proposed architecture in Yuan et al. to avoid collisions when the self-driving vehicle detects other car first in the collision area. Similarly, ML-REQ2 and ML-REQ3 tackles the fact that the vehicle trained on the proposed algorithm, will not attempt to pass the intersection first if it is assigned a lower-level policy compared to the other cars in that particular scenario.
We further define the ML performance and robustness requirements as explained in [2, 3].
**Performance requirements**: The performance requirements correspond to how accurately the self-driving vehicle trained on the D3QNPER algorithm combined with the level-k reasoning, avoids collision. We derived the following requirements to fulfill the above statement

1. **PER-REQ1**: The ego vehicle with the adaptive policy avoids collision with a high success rate of at least 96%.
2. **PER-REQ2**: The ego vehicle with the adaptive policy avoids collision with a success rate of at least 2% better than AV controller [119].

**Robustness requirements**: These requirements make sure that the proposed architecture performs well in spite of slight variations in input.

1. **ROB-REQ1**: The ego vehicle with the adaptive policy shall perform as expected in all the situations which the ego vehicle may encounter within the defined operational environment.

2. **ROB-REQ2**: The self-driving vehicle shall start deceleration action irrespective of the higher-level agents as soon as it detects the other car in the collision area first

**ARTEFACT [J]: Validate ML safety requirements** **SIMULATION RESULTS**: Section C-hardware implementation of Yuan et al. [1] provides the simulation results for four scenarios of the unsignalized 4-way intersection. In the simulation results, it can be seen that when the vehicles interact with other vehicles assigned with different policies, collision is prevented and all of them pass the intersection safely.

**ARTEFACT [I]: ML safety requirements argument pattern:** Figure 26 represents the ML safety requirements argument [I] adapted from [3] and modified according to our system. We established the main objective of this safety argument, which is to make sure that the requirements allocated to ML component are satisfied in the development of the ML model by using G2.1. We supported this goal with an argument S2.1 that is split into four sub-goals. The first three sub-goals state that the D3QN PER algorithm combined with level-k reasoning as proposed in Yuan et al. [1] indeed satisfies the ML safety requirements. The fourth sub-goal G2.5 makes sure that these ML safety requirements are valid, which is supported by the evidence Sn1 (artefact [J] explained above). Finally, through the use of argument S2.2 and sub-goals G2.6 and G2.7 we demonstrate that the performance and robustness requirements are satisfied. This is backed up by the hardware implementation results presented in [1] as evidence Sn2.

### 6.3.3 Stage 6 – Model Deployment

The main objective in this stage is to show that the system satisfies its allocated safety requirements after the deployment of the developed model and also to create a safety argument that demonstrates that the ML model will continue to satisfy the safety requirements even after its integration into the system.

**ARTEFACT [V] : ML model** Yuan et al. [1] developed a combination of D3QN PER algorithm with level-k reasoning in order for the self-driving vehicles to cross the unsignalized 4-way intersection safely and effectively without any collision.

**ARTEFACT [EE] : Operational scenarios** :
According to the definition of operational situation presented in [5], it is defined as the *"scenario that could occur throughout the lifecycle of the vehicle"*. According to ISO 21448, it is defined as *"Description of an imagined sequence of events that includes the interaction of the product or service with its environment and users, as well as interaction among its product or service components"*. Yuan et al. [1] describes the following operational scenarios in their hardware implementation results (simulation):

1. **Scenario -1** : car1 controlled by level-1 policy (conservative) and rest three cars controlled by level-2 policy (aggressive)

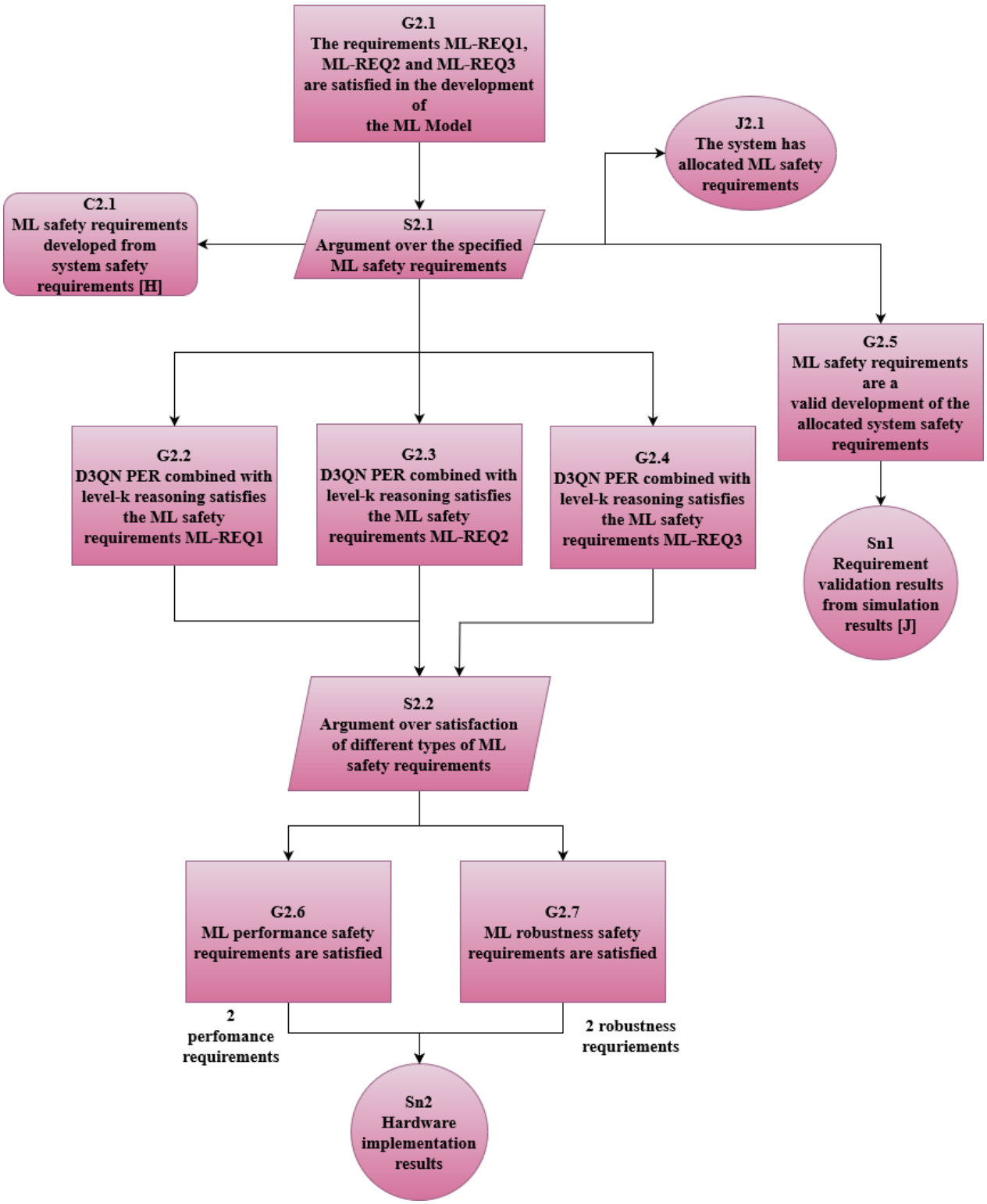


Figure 26: ML safety requirements argument [I] adapted from [3] and modified according to our system

2. **Scenario -2**: car1 and car4 controlled by level-1 policy, car2 and car3 controlled by level-2 policy
3. **Scenario -3**: car 2 controlled by level-2 policy interacts with others controlled by level-1 policy with car 1 turning left
4. **Scenario -4**: car 1 turning right controlled by level-2 interacts with others controlled by level-1

**ARTEFACT [FF] : Integration testing results** After training the model and integrating it into the ego vehicle, based on the simulation results presented in Section C of Yuan et al. [1], we can see that these trained models obtained from the simulator, is able to deal with complex interaction scenarios in the unsignalized 4-way intersection, which in turn verifies the feasibility of the proposed approach in Yuan et al. [1]

**ARTEFACT [GG]: ML deployment argument pattern** Figure 27 represents the ML deployment argument. The main objective of this safety argument is to demonstrate that the ML safety requirements are still satisfied even after integration of the ML model into the target system. This is denoted by the goal G6.1. This goal is supported by an argument S2.1 which is then split into two sub-goals. Additionally, we supported the argument S2.1 with the context C6.1 of operational scenarios [EE] and a justification J6.1 in which we state that these scenarios chosen in [1] represents complex scenarios. Through the use of sub-goals G6.2, G6.3, G6.4, G6.5, and G6.6, we demonstrate that the ML safety requirements and system safety requirements are satisfied throughout the system operation. We back up these goals through the use of evidences Sn1 and Sn2 which are the artefacts simulation results [J] and integration test results [FF] described above.

## 6.4 Lessons learnt

In this section, we share the most valuable lessons that we learnt when using our framework to create a safety case.

### 6.4.1 Construction of safety case

We constructed the safety case presented in this thesis by following and extending the guidelines presented in AMLAS [3] and also adapting the safety case presented in SMIRK [2] which followed AMLAS. As discussed previously, we know that AMLAS is applied during the development of the ML component. In SMIRK, they follow similar format and applied AMLAS during the development of the pedestrian detection component. Additionaly, the authors of SMIRK also claim that AMLAS can be successfully applied after initial prototyping, which even accelerated the safety assurance process. However, there is a considerable difference between the design approach we propose in this thesis and that of AMLAS and SMIRK. In our thesis, we created a safety case of a system [1] that is already developed, trained, tested and successfully running. Hence, we selected the stages of AMLAS that was applicable for our system and crafted a framework based on this information. Selecting the applicable stages of AMLAS was a crucial process as we carefully went through the AMLAS and SMIRK documentation multiple times to make sure we do not miss any important stages. Hence, we can confidently say that we can indeed use AMLAS as a foundation for creating

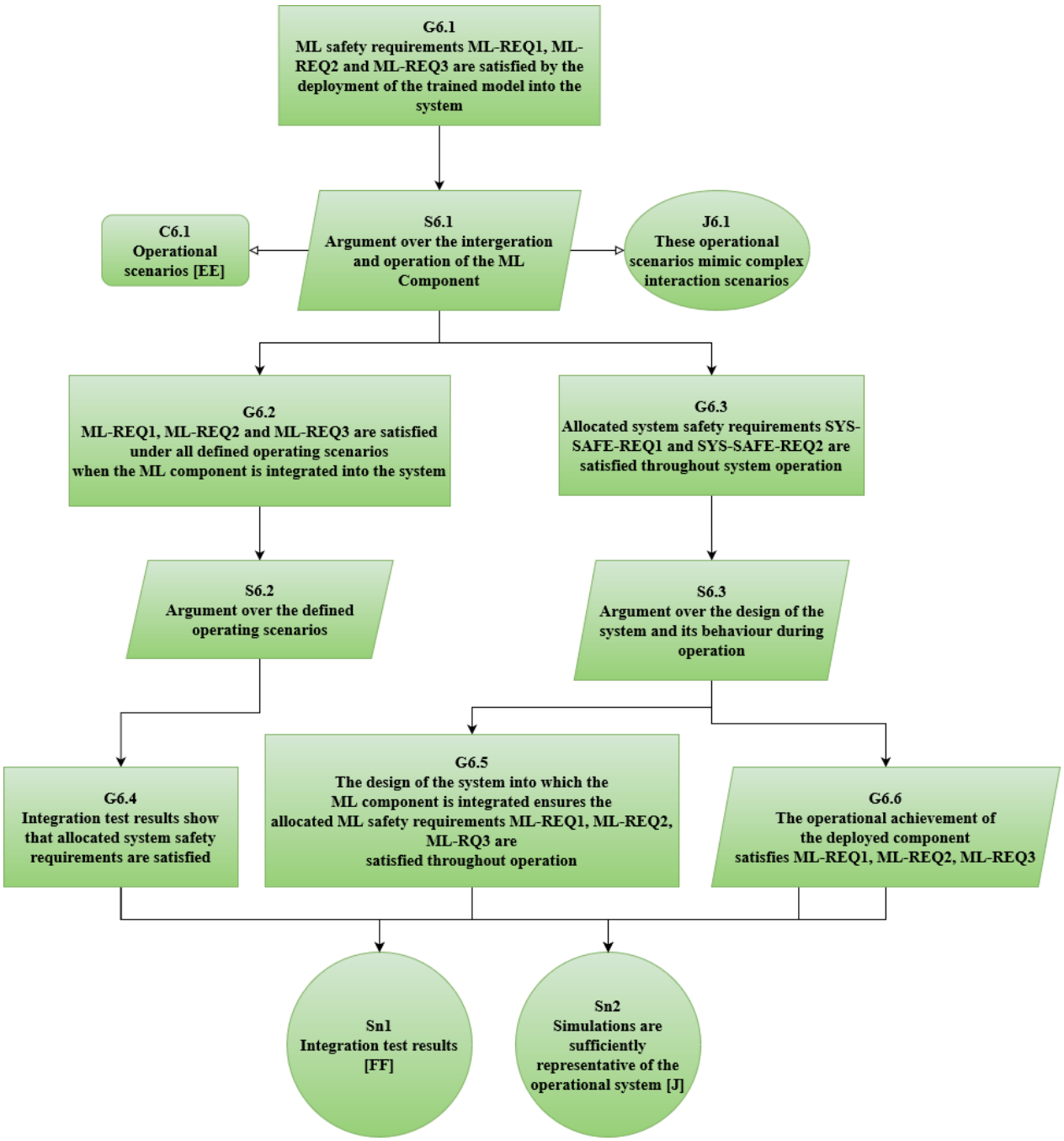


Figure 27: ML deployment argument pattern [GG] adapted from [3] and modified according to our system

safety case for an already developed ML component. In fact, we can even extend and modify AMLAS to support dynamic safety assurance similar to our case (a third change impact analysis phase).
Dynamic safety assurance refers to the continuous monitoring, assessment, and adaptation of safety measures in real-time or over time in response to changing conditions, requirements, or risks within a system or environment [120, 121, 100, 122, 123]. Unlike static safety assurance approaches, which typically rely on pre-defined safety cases or analyses conducted at specific points in time, dynamic safety assurance is inherently adaptive and responsive to evolving circumstances. Approaches like AMLAS primarily support static safety assurance, focusing on the initial development and validation of safety cases or assurance arguments for autonomous systems [3]. While static safety assurance is essential for establishing baseline safety requirements and confidence in system behavior, dynamic safety assurance complements these efforts by enabling ongoing monitoring, adaptation, and verification of safety-critical properties throughout the system's lifecycle.

### 6.4.2 Extending the application of the design approach to other scenarios

In this paper, we successfully applied AMLAS to an unsignalized 4-way intersection scenario which focuses on avoiding collisions between cars [1]. This scenario is quite different from SMIRK which focuses on a pedestrian identification component. Since, we were able to follow SMIRK and AMLAS and created a safety case for a completely different scenario and ML algorithm, we think that the same design approach can be adapted to different scenarios such as lane change, merging into highway, parking maneuver, obstacle avoidance, emergency vehicle approach etc. For example, the recent incident of general motors cruise over wet cement [8] can also be considered as a scenario for which our design approach can be adapted.

[8] https://www.digitaltrends.com/cars/an-autonomous-car-in-san-francisco-got-stuck-in-wet-concrete/

# 7 Results of RQ3: automatic generation of safety cases using GPT-4

## 7.1 RQ3.1: Is GPT-4 Adequately Skilled in Goal Structuring Notation?

Our evaluation process encompassed two distinct types of assessments. In the first assessment, where we had two graduate students (assessors/raters) grade the responses to the 19 questions posed to GPT, we aimed to quantify the inter-rater agreement using the Kendall's Tau correlation. The results of this analysis are presented in Table 6. These results indicate the average correlation between the scores assigned by the two raters is **0.73**. This high level of correlation signifies a strong agreement between the two raters assessments. An intriguing observation drawn from the outcomes of RQ3.1 is that GPT-4 exhibited a strong performance in syntactic rule-based questions. However, in the case of structural rule-based questions, GPT-4's performance lagged behind the other two question categories. Nevertheless, GPT-4 demonstrated a reasonably good performance in generation-based questions.

Table 6: Kendall's Tau correlation values for four RQ1 runs and their average

| Run-1 | Run-2 | Run-3 | Run-4 | Average |
|---|---|---|---|---|
| 0.66 | 0.86 | 0.63 | 0.77 | 0.73 |

Table 7: Average scores for all RQ1 questions

| Assessor | Run-1 | Run-2 | Run-3 | Run-4 | Average |
|---|---|---|---|---|---|
| 1 | 1.37 | 1.63 | 1.53 | 1.37 | 1.48 |
| 2 | 1.16 | 1.58 | 1.42 | 1.47 | 1.40 |

Secondly, we aimed to evaluate GPT-4's competence in responding to fundamental queries related to the Goal Structuring Notation (GSN). The results of this assessment, including the average scores from the four rounds of prompting GPT-4 to answer GSN-related questions, are presented in Table 7. Analysis of these results reveals that the average score across all four rounds for both the assessors is 1.44 (average of 1.48 and 1.40), indicative of GPT-4's robust understanding of the fundamental aspects of GSN. According to the grading scheme utilized by several universities, this performance corresponds to a **grade of A** i.e. an excellent grade. It is noteworthy that GPT-4 excelled in answering questions related to syntactic correctness. Besides, GPT-4 is able to correctly specify how most of the GSN elements should be depicted and connected by relationships. GPT-4 is therefore able to generate safety cases in a hierarchical format, mirroring the structure of a goal hierarchy and resembling structured prose. In most of the rounds, it is also able to correctly generate samples of GSN elements.

The box below highlights a correct answer provided by GPT-4 regarding GSN. It emphasizes one of the key attributes of GSN, which is its nature as a directed acyclic graph (DAG), meaning there are no cycles within the GSN diagram. GPT-4 accurately recognizes this characteristic. Furthermore, it is capable of elucidating the goal structure.

**Qn: What type of graph is a goal-structure in GSN ?**
A goal-structure in GSN is a **directed acyclic graph (DAG)**, where nodes represent goals, strategies, evidence, etc., and edges indicate relationships.

In the below inquiry, it is important to note that GPT-4 provided incorrect responses in relation to assumptions and justifications, which should have been represented using ellipses and not ”angled brackets” or ”square brackets” as GPT-4 suggested. The textbox below illustrates that incorrect answer (highlighted in italic bold) that GPT-4 generated during our experiments.

**Qn: Can you explain how each of the six elements of GSN are structurally represented. i.e., what shapes?**
Goals: Represented by rectangles. Strategies: Illustrated using parallelograms. Solutions: Shown as circles or ovals. Contexts: Depicted with rounded rectangles. ***Assumptions: Represented by text in angled brackets. Justifications: Illustrated as text in square brackets.***

This indicates a limitation of GPT-4's understanding of the GSN syntax (notation) related to these specific two elements. Still, adopting the structured prose complying with the GSN instead of a visual representation (e.g., graphical goal structures created with GSN) prevents that limitation from having an impact on the GSN elements generated by GPT-4. To represent safety cases, we therefore adopt the structured prose complying with GSN in the remainder of this thesis.

In summary, for **RQ3.1**, our conclusion is that GPT-4 has demonstrated a solid grasp of fundamental GSN knowledge, and it successfully passed this assessment with a **grade of A i.e. with an excellent grade.**.

## 7.2 RQ3.2: Is GPT-4 Able to Generate Comprehensive and Accurate Safety Cases?

### 7.2.1 Ground-truth similarity

**Sequence matching scores:** The mean sequence matching scores (refer to Table 8) across all experiments indicate an average value of 0.049. The low averages across all three safety cases suggest that the safety cases generated by GPT-4 differ significantly from the ground-truth ones in terms of textual similarity. This divergence may stem from GPT-4's utilization of a distinct vocabulary compared to that of safety case modelers. Furthermore, it's important to acknowledge that no two safety cases are identical; variations in presentation style and format are common among different creators. Consequently, multiple versions of the same safety case may exist. This result underscores this variability.

Table 8: Sequence Matching scores

| Safety case | Exp-1 | Exp-2 | Exp-3 | Exp-4 | Average |
|---|---|---|---|---|---|
| ML | 0.03125 | 0.0555 | 0.05225 | 0.0535 | 0.048125 |
| X-ray | 0.036 | 0.072 | 0.054 | 0.1205 | 0.070625 |
| LMS | 0.03075 | 0.0295 | 0.02675 | 0.0285 | 0.028875 |
| Total Avg | | | | | 0.049 |

**Bleu scores:** The mean sequence matching scores (refer to Table 9) across all experiments indicate an average value of 0.017. It can be seen that this score is even lower than the average sequence matching

scores. As explained in the previous section, this discrepancy may arise from GPT-4's utilization of a distinct vocabulary and the inherent variability among safety cases. Indeed, it's widely recognized that no two safety cases are identical, further accentuating the potential for variations in language and structure.

Table 9: Bleu scores

| Safety case | Exp-1 | Exp-2 | Exp-3 | Exp-4 | Average |
|---|---|---|---|---|---|
| ML | 0 | 0.0115 | 0.0105 | 0 | 0.0055 |
| X-ray | 0 | 0.06075 | 0.0055 | 0.0695 | 0.0339375 |
| LMS | 0.004 | 0.014 | 0.006 | 0.02575 | 0.0124375 |
| Total Avg | | | | | 0.017 |

**Cosine (semantic) similarity** When examining the average cosine similarity scores that Table 10 reports, we find that for the ML safety case across all four experiments, the score is 0.89, for the X-ray safety case, it is 0.87 and for the LMS safety case, it is 0.86. These cosine similarity values are very close to 1, which indicates a high level of similarity between the ground-truths and GPT-4 generated safety cases. From the perspective of cosine similarity, we can infer that GPT-4 successfully generated safety cases that are semantically similar to the original ground-truth safety cases.

Table 10: Cosine similarity scores

| Safety Case | Exp-1 | Exp-2 | Exp-3 | Exp-4 | Avg |
|---|---|---|---|---|---|
| ML | 0.9 | 0.88 | 0.897 | 0.895 | 0.89 |
| X-ray | 0.867 | 0.902 | 0.817 | 0.895 | 0.87 |
| LMS | 0.78 | 0.92 | 0.81 | 0.91 | 0.86 |

### 7.2.2 Reasonability

The average reasonability scores (see Table 11), encompassing all four experiments, yield an average of 2.69. When rounded to the nearest whole number, this results in a score of 3. In the context of the ML safety case, this indicates that GPT-4's generated safety case elements are moderately reasonable. In the case of LMS safety case, the average reasonability score is 2, indicating that GPT-4 was able to generate mostly reasonable safety arguments. For the X-ray safety case, the average reasonability score is 2.27, rounding off to a score of 2. That score indicates that GPT-4 was able to generate mostly reasonable safety arguments for that safety case. Noteworthy, Figure 28 showcases one of the safety cases generated by GPT-4 for the LMS system, for run-1 of experiment-4. GPT-4 generated this safety case in the structured prose format. We then converted that text based safety case into GSN format and relied on Figure 28 to illustrate the resulting goal structure.

These findings suggest that GPT-4 demonstrates the capability to generate coherent safety case elements. While there may be instances where certain elements differ from the actual ground-truth safety case, they are generally considered to be moderately or mostly reasonable, depending on the specific safety case context. This underscores GPT-4's ability to generate safety case components that exhibit a reasonable level of

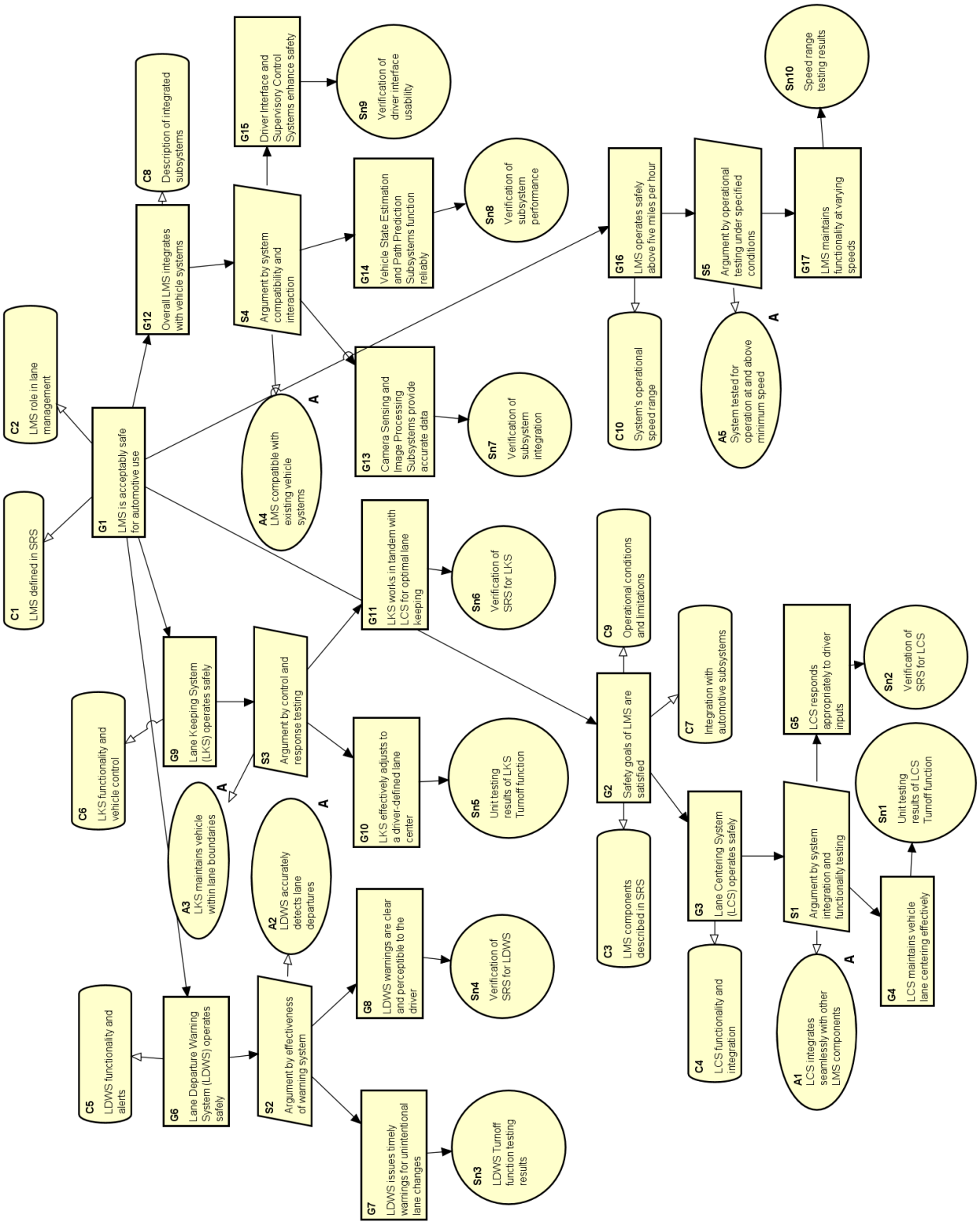


Figure 28: Safety case generated by GPT-4 for run-1 of experiment-4 of the LMS safety case, depicted in GSN format

Table 11: Reasonability assessment scores

| Safety Case | Exp-1 | Exp-2 | Exp-3 | Exp-4 | Average |
|---|---|---|---|---|---|
| ML | 2.75 | 2.75 | 2.5 | 2.75 | 2.69 |
| X-ray | 2.5 | 2 | 2.25 | 2.5 | 2.31 |
| LMS | 2.7 | 2.4 | 2.8 | 1.9 | 2 |

coherence and logic. This implies that GPT-4 is able to generate GSN elements that the safety case modeler (argument developer) may not have thought of, but that can still be relevant for the modeled safety case. Hence, GPT-4 has the ability to generate GSN elements that can enrich the safety cases being modeled by a safety case modeler.

### 7.2.3 Analysis of the results of RQ3

An interesting observation arises when we analyze these three measures (i.e. sequence matching scores, bleu scores, reasonability and semantic similarity) together. According to sequence matching and bleu scores (see Table 8 and Table 9), GPT-4 performs very poorly. This may be attributed to the fact that GPT-4 uses a different vocabulary compared to safety case modelers. Also, there can exist different versions of the same safety case. So, this does not undermine the safety case generated by GPT-4. It is just completely textually different from the actual ground-truth safety case. To improve the results of the sequence matching scores and the bleu scores, it may be interesting to explore in future work the possibility to force GPT-4 to adopt the same controlled vocabulary that safety case developers usually use when manually creating safety cases. Furthermore, the values of the semantic similarity measure is significantly better than the ones of the sequence matching scores and bleu scores. This contrast may stem from the fact that the computation of cosine similarity –that is used to compute the semantic similarity – primarily considers the semantic resemblance between two safety cases, without directly assessing their textual correctness. Therefore, while GPT-4's generated safety cases that exhibit semantic similarity to the reference cases, there may still be areas where their correctness or reasonability can be improved. These observations highlight the importance of considering multiple evaluation measures to comprehensively assess the quality of the generated safety cases.
Upon analyzing the outcomes of the four experiments, it becomes evident that GPT-4 often excels in generating safety cases that closely mirror the original safety case in experiments 2 and 4. This achievement can primarily be attributed to the distinctive characteristics of these experiments. In these experiments, we furnish GPT-4 with the domain context. In addition, in experiment-4 we also provide GPT-4 with the GSN syntax which further enhances GPT-4's performance. As also discussed in [29], GPT-4 exhibits enhanced performance when supplied with comprehensive contextual information and a clear understanding of the syntax.

| In summary, our findings for **RQ3.2** indicate that GPT-4 possesses the capability to generate safety cases that are at least moderately reasonable. Additionally, GPT-4 generated safety cases exhibit a high degree of semantic similarity with the ground-truth safety cases. |
|---|

# 8 Threats to validity

Based on the framework that Wohlin et al. [124] and Zhou et al. [125] proposed, we have identified the following threats to validity.

## 8.1 Threats for RQ1:

### 8.1.1 Internal validity

Several bibliometric analyses select studies based on the first three phases of our selection strategy. This usually yields several hundred/thousands of studies and makes the selection process less time-consuming. This may also yield a relatively high amount of noise in the so-selected studies. We, therefore, built on the features of common secondary studies guidelines (e.g., PRISMA 2020 and SEGRESS) to propose a more rigorous selection strategy that consists of five phases. This had the advantage of significantly reducing the noise in the selected studies by allowing the selection of studies that are more relevant to the research topic. However, this also significantly reduced the number of primary studies included in our bibliometric analysis and therefore reduced the scope of our bibliometric analysis. Nonetheless, our bibliometric analysis analyses several hundred of studies and can therefore serve as a reference to position future work on the safety assurance topic.

### 8.1.2 Conclusion validity

The field of safety assurance is inherently interdisciplinary, demanding expertise in many domains including those we explored in our analysis. A lack of such interdisciplinary expertise could result in an incomplete analysis that may not provide enough insights into the surveyed topic. To mitigate that issue, our research team is constituted with members with strong expertise in computer science (with a focus on software engineering and machine learning), along with extensive experience in automotive and aerospace.

## 8.2 Threats for RQ2:

### 8.2.1 Construction validity:

For the system introduced by Yuan et al. [1], we developed the safety case by specifically opting for stages 1, 2, and 6 from the AMLAS framework [3]. While we have rationalized that these stages adequately address our needs, there remains a possibility that other stages from AMLAS could also be relevant. To ensure the thoroughness of our selection, we meticulously reviewed the chosen stages. Based on our assessment, it is highly improbable that any additional stages from the AMLAS framework would offer relevance to our specific case.

## 8.3 Threats for RQ3:

### 8.3.1 Construction validity

We derived the contextual information incorporated into the prompts for the experiments conducted in response to RQ3.2, from the details provided in the corresponding references [6, 7, 8] about the systems for which the ground-truth safety cases were created. Hence, our knowledge of these systems is confined to what has been explicitly presented within those references. Any additional or nuanced information pertaining to these systems and that might exist beyond the scope of those papers is not accessible to us. Therefore, it is essential to acknowledge that the safety cases generated by GPT-4 are solely based on the information available within the referenced papers. There is a possibility that critical or contextually significant details about the system may have been omitted or not explicitly addressed in those references.

### 8.3.2 Conclusion validity

The version of GPT-4 we utilized is known to be non-deterministic, which means that running the model multiple times can yield slightly different results. To ensure the reliability and consistency of our experiments in the future, we will explore the use of a deterministic version of GPT. This deterministic variant may provide stable and predictable outcomes, eliminating the variability associated with non-deterministic models.

# 9 Discussion and Future work

## 9.1 Develop advanced solutions to foster regulatory safety compliance

Figure 16 highlights that the majority of primary studies in our bibliometric analysis mainly focus on safety assurance standards in the Automotive (e.g., ISO 26262), Aerospace (e.g., ARP 4754-A and DO-178C) and Railway (e.g., EN 501X) domains. However, a notable gap exists in specific standards that explicitly mandate safety assurance for modern AI-enabled cyber-physical systems. This gap is concerning given the increasing prevalence of AI in safety-critical domains. Collaboration between AI companies and governments, as Sam Altman suggests [126], could lead to the creation of a regulatory agency to oversee AI system development and safety testing. Implementing AI regulations poses challenges such as balancing national and global standards and addressing tech giants' concerns about staying competitive. Standards like UL 4600 [127] for autonomous vehicles and safety cases are essential for ensuring compliance and enhancing accountability in AI-enabled systems.

## 9.2 Integrate Safety Assurance into Agile/DevOps: Aligning Safety Case Development with Agile Methodologies

Figure 16 shows that there has been relatively limited research dedicated to agile safety cases. The rationale for this observation could be attributed to the fact that agile safety cases are considered an emerging field within the broader domain of safety assurance. The concept of agile safety cases is relatively new and may not have gained as much traction as more traditional approaches in safety case development. While significant contributions have been made in integrating agile and DevOps practices into safety-critical systems, as demonstrated by notable examples (e.g., [128, 129, 130], it remains obvious that the adoption of agile methodologies within the research domain of safety cases is not yet widespread. Despite advancements in other areas of safety-critical systems, the incorporation of agile approaches into the specific context of safety cases remains a relatively unexplored and underutilized avenue. However, in recent years, as shown in Figure 17, there has been a noticeable increase in the adoption of agile development methods for safety-critical software. This shift has been driven by the desire to streamline the software development process, reduce costs, and improve overall software quality. Embracing an agile approach allows safety cases to be developed incrementally as information becomes available during the project, leading to greater safety awareness and understanding throughout the development life cycle [131, 132, 133].

## 9.3 Provide Dynamic Assurance to ensure the safety of Cyber-Physical Systems (CPSs): From Development to Life Cycle Management

Emerging technologies, such as CPSs (e.g., ML-enabled autonomous driving systems, and unmanned aerial systems) –that are usually self-adaptive and self-managing systems – call for a paradigm shift in safety assurance. More specifically, our analysis from Figure 16 allows us to conclude that traditional approaches to safety certification and assurance are not sufficient to address the dynamic nature of these systems. There is therefore a pressing need for a new class of safety certification techniques that continuously assess and

evolve safety reasoning, aligning with the system's life cycle.
Furthermore, the integration of machine learning techniques holds immense potential for enhancing the efficiency and accuracy of safety assurance processes in these systems [134]. Additionally, interdisciplinary collaboration between experts in software engineering, machine learning, and domain-specific fields is essential to ensure a holistic approach to safety assurance in self-adaptive systems. This approach aims to provide through-life safety assurance, encompassing not only the initial development and deployment stages but also run-time monitoring based on real-world operational data [120]. To ensure safety and reliability in modern complex systems, especially self-adaptive systems, dynamic safety assurance, as a continuous and adaptable process, becomes essential. This area holds significant promise for future research and development in safety assurance
Neto et al. [84] further echo the need to support the dynamic safety assurance of intelligent safety-critical systems.

## 9.4 Automate the generation of safety cases

The generation of safety cases is predominantly a manual process, which poses several challenges. The manual approach is not only expensive but also prone to errors, and it demands significant labor and time investment. One of the primary hurdles lies in constructing and evaluating safety cases, given the substantial amount of information required to demonstrate adherence to relevant regulatory standards such as ISO 26262 and DO-178C. The extensive and diverse nature of information needed in safety cases makes the process complex and difficult to manage effectively. Hence, one of the key future directions for researchers to focus on the field of SAC is the automation of safety case generation. Automating the process of creating safety cases can significantly improve efficiency and reduce manual effort [134].

## 9.5 GPT-4's contribution to streamlining argumentation for safety case modelers

Our experiments show GPT-4 is able to generate safety arguments that are at least moderately accurate and reasonable when provided with relevant information. These moderate generation abilities led us to the conclusion that human intervention is still essential in the safety case generation process. There are two possible avenues for this intervention. The first one is incremental refinement, as established in [24]. Human involvement can take the form of incremental guidance, where a human interacts with GPT-4 to refine and shape the generated safety arguments. This iterative process ensures that the arguments align more closely with the specific system in question and adhere to the desired safety standards. The second one is manual expert analysis, where, a human safety expert can conduct a thorough manual analysis of the safety arguments generated by GPT-4. This expert can verify that all relevant information is captured and may make necessary changes to tailor the safety arguments to the particular system, thereby ensuring its safe deployment.

## 9.6 GPT-4 vs. the safety case modeler: A replacement debate

Our initial findings indicate that the current state of automation in safety case generation with GPT-4 presents certain limitations and challenges that must be acknowledged. Safety-critical systems, known for their potential catastrophic consequences in case of failure, call for the creation of highly reliable and accurate safety cases to demonstrate how they meet safety expectations. Therefore, based on our results, the idea of fully automating safety case generation, without human involvement, may not be practical or advisable at this stage. A significant concern is that GPT-4's safety cases may achieve, at best, approximately 80-90 percent structural and semantic correctness. This level of accuracy may not meet the rigorous standards required for safety-critical systems, where precision is paramount. Hence, involving safety case modelers in the safety case development process remains essential, at least for now. These experts possess domain-specific knowledge, critical thinking skills, and years of expertise, ensuring not only structural and semantic correctness but also the incorporation of crucial domain-specific insights into safety cases. Therefore, human intervention is indispensable in the safety case generation process to ensure the highest quality and precision. Still, in the future, the advent of more advanced large language models or other generative AI-based solutions may undermine our conclusions as they may allow the automatic generation of safety cases exhibiting outstanding structural and semantic correctness.

# 10 Conclusion

Our bibliometric analysis harnesses common bibliometric tools, to quantitatively analyze primary studies drawn from well-known databases and focusing on safety assurance cases. This allows examining the trends in publication years, and the top venues where the primary studies were published. This also allows exploring the most influential keywords in the field of safety cases, the most cited primary studies, and the diversity of affiliations of authors. This allows identifying future directions in the SAC research. As future work, we aim at building upon the foundations laid by this study by shifting our focus towards a qualitative synthesis of the data extracted from primary studies. This will allow getting additional insights on the topic under investigation.

Furthermore, we have manually created a safety case by extending on AMLAS [3] and SMIRK [2], for a deep learning based reinforcement learning algorithm designed to avoid collisions on an unsignalized 4-way intersection scenario presented of an autonomous vehicle presented in [1]. We carefully select the stages of AMLAS that will be relevant for our case and meticulously crafted the methodology for creating safety case. We have also presented the complete safety case (including 10 artefacts and 3 safety arguments) for the target system mentioned above.

Additionally, we have delved into the potential of harnessing generative AI for the automated creation of safety cases. Our research has yielded several noteworthy findings:(a) GPT-4 has demonstrated a commendable proficiency in comprehending and effectively utilizing the goal structuring notation (GSN). To be more specific, in our rigorous evaluation, GPT-4 has displayed a remarkable level of competence, achieving an excellent grade of **A** in its ability to work with GSN. (b) Furthermore, our experiments reveal that GPT-4 possesses the capability to generate safety cases that exhibit at least a moderate level of semantic accuracy and reasonability. More specifically, it excels in generating safety arguments that bear a striking semantic similarity to the ground-truth safety cases. In summary, from dissecting publication trends to crafting intricate safety arguments, we've paved the way for future research to delve deeper into the intricate world of safety assurance cases.

# Appendices

## A List of all primary studies

Table 12: List of Primary Studies

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P1 | Dezfuli, H et al. [135] | 2012 | A ”Systems/Case-Based” approach to system safety | ESREL and PSAM |
| P2 | Cassano, V et al. [136] | 2016 | A (Proto) logical basis for the notion of a structured argument in a safety case | ICFEM |
| P3 | Denney, E et al. [137] | 2013 | A formal basis for safety case patterns | SAFECOMP |
| P4 | Wang, R et al. [138] | 2016 | A framework for assessing safety argumentation confidence | SERENE |
| P5 | Hawkins, R et al. [139] | 2012 | A framework for determining the sufficiency of software safety assurance | SSCS |
| P6 | Despotou, G et al. [140] | 2017 | A framework for synthesis of safety justification for digitally enabled healthcare services | Digital Health |
| P7 | Lin, C et al. [141] | 2016 | A Framework to Support Generation and Maintenance of an Assurance Case | ISSREW |
| P8 | Yamamoto, S | 2014 [142] | A knowledge integration approach of safety-critical software development and operation based on the method architecture | KES |
| P9 | Birch, J et al. [143] | 2014 | A layered model for structuring automotive safety arguments | EDCC |
| P10 | Viger, T et al. [144] | 2021 | A Lean Approach to Building Valid Model-Based Safety Arguments | MODELS |
| P11 | Denney, E et al. [145] | 2012 | A lightweight methodology for safety case assembly | SAFECOMP |
| P12 | Björnander, S et al. [146] | 2012 | A method to formally evaluate safety case arguments against a system architecture model | ISSREW |
| P13 | vSljivo, I et al. [147] | 2017 | A method to generate reusable safety case argument-fragments from compositional safety analysis | JSS |
| P14 | Guiochet, J et al. [148] | 2015 | A model for safety case confidence assessment | SAFECOMP |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P15 | Gallina, B [149] | 2014 | A model-driven safety certification method for process compliance | ISSREW |
| P16 | Larrucea, A et al. [150] | 2015 | A modular safety case for an IEC-61508 compliant generic COTS multi-core processor | CIT |
| P17 | Larrucea, A et al. [151] | 2015 | A modular safety case for an IEC-61508 compliant generic hypervisor | DSD |
| P18 | Khalil, M et al. [152] | 2014 | A pattern-based approach towards the guided reuse of safety mechanisms in the automotive domain | IMBSA |
| P19 | Nevsić, D et al. [153] | 2021 | A probabilistic model of belief in safety cases | Safety Science |
| P20 | Idmessaoud, Y et al. [154] | 2022 | A Qualitative Counterpart of Belief Functions with Application to Uncertainty Propagation in Safety Cases | BELIEF |
| P21 | Feng, L et al. [155] | 2014 | A safety argument strategy for PCA closed-loop systems: A preliminary proposal | MCPS |
| P22 | Schmid, T et al. [156] | 2019 | A Safety Argumentation for Fail-Operational Automotive Systems in Compliance with ISO 26262 | ICSRS |
| P23 | Ayoub, A et al. [157] | 2012 | A safety case pattern for model-based development approach | NFM |
| P24 | Liu, Q et al. [158] | 2012 | A safety-argument based method to predict system failure | PHM |
| P25 | Menon, C et al. [159] | 2020 | A safety-case approach to the ethics of autonomous vehicles | Safety and Reliability |
| P26 | Birch, J et al. [160] | 2020 | A Structured Argument for Assuring Safety of the Intended Functionality (SOTIF) | WAISE |
| P27 | Luo, Y et al. [161] | 2017 | A systematic approach and tool support for GSN-based safety case assessment | JSA |
| P28 | Ayoub, A et al. [162] | 2012 | A systematic approach to justifying sufficient confidence in software safety arguments | SAFECOMP |
| P29 | Vorapojpisut, S [163] | 2016 | A V-model framework for the certification against the Annex R of IEC 60335-1: Class B appliances | ICIT |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P30 | Denney, E et al. [24] | 2012 | AdvoCATE: An assurance case automation toolset | SASSUR |
| P31 | Myklebust, T et al. [132] | 2020 | Agile safety case and DevOps for the automotive industry | ESREL and PSAM |
| P32 | Myklebust, T et al. [133] | 2022 | Agile safety case for vehicle trial operations | PSAM |
| P33 | de la Vara, J et al. [164] | 2017 | An analysis of safety evidence management with the Structured Assurance Case Metamodel | CSI |
| P34 | Ward, F et al. [165] | 2020 | An Assurance Case Pattern for the Interpretability of Machine Learning in Safety-Critical Systems | DECSoS |
| P35 | Yamamoto, S et al. [166] | 2013 | An evaluation of argument patterns to reduce pitfalls of applying assurance case | ASSURE |
| P36 | Nair, S et al. [167] | 2016 | An evidential reasoning approach for assessing confidence in safety evidence | ISSRE |
| P37 | Matsuno, Y et al. [165] | 2013 | An implementation of GSN community standard | ASSURE |
| P38 | Larrucea, X et al. [168] | 2018 | Analyzing a ROS based architecture for its cross reuse in ISO26262 settings | MEDI |
| P39 | Lin, C et al. [169] | 2015 | Applying Safety Case Pattern to Generate Assurance Cases for Safety-Critical Systems | HASE |
| P40 | Gleirscher, M et al. [170] | 2017 | Arguing from hazard analysis in safety cases: A modular argument pattern | HASE |
| P41 | Cârlan, C et al. [171] | 2017 | Arguing on software-level verification techniques appropriateness | SAFECOMP |
| P42 | Hocking, A et al. [172] | 2014 | Arguing software compliance with ISO 26262 | ISSREW |
| P43 | Grigorova, S et al. [173] | 2014 | Argument Evaluation in the Context of Assurance Case Confidence Modeling | ISSREW |
| P44 | Denney, E et al. [174] | 2015 | Argument-based airworthiness assurance of small UAS | DASC |
| P45 | Yuan, T et al. [175] | 2012 | Argument-based approach to computer system safety engineering | IJCCBS |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P46 | Reich, J et al. [176] | 2020 | Argument-Driven Safety Engineering of a Generic Infusion Pump with Digital Dependability Identities | IMBSA |
| P47 | de la Vara, J et al. [177] | 2019 | Assessment of the Quality of Safety Cases: A Research Preview | REFSQ |
| P48 | Picardi, C et al. [178] | 2020 | Assurance argument patterns and processes for machine learning in safety-related systems | SafeAI |
| P49 | Y. Zhang et al. [179] | 2018 | Assurance case considerations for interoperable medical systems | ASSURE |
| P50 | Schwierz, A et al. [180] | 2018 | Assurance Case to Structure COTS Hardware Component Assurance for Safety-Critical Avionics | DASC |
| P51 | Asaadi, E et al. [181] | 2020 | Assured Integration of Machine Learning-based Autonomy on Aviation Platforms | DASC |
| P52 | Denney, E et al. [182] | 2014 | Assuring ground-based detect and avoid for UAS operations | DASC |
| P53 | Conmy, P et al. [183] | 2014 | Assuring safety for component based software engineering | HASE |
| P54 | Reich, J et al. [184] | 2019 | Automated evidence analysis of safety arguments using digital dependability identities | SAFECOMP |
| P55 | Hartsell, C et al. [185] | 2021 | Automated Method for Assurance Case Construction from System Design Models | ICSRS |
| P56 | Armengaud, E [186] | 2014 | Automated safety case compilation for product-based argumentation | ERTS |
| P57 | Cârlan, C et al. [187] | 2022 | Automating Safety Argument Change Impact Analysis for Machine Learning Components | PRDC |
| P58 | Denney, E et al. [188] | 2014 | Automating the assembly of aviation safety cases | TOR |
| P59 | Macher, G et al. [189] | 2014 | Automotive safety case pattern | EuroPLoP |
| P60 | Idmessaoud, Y et al. [190] | 2020 | Belief Functions for Safety Arguments Confidence Estimation: A Comparative Study | SUM |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P61 | Williams, B et al. [191] | 2014 | Building the safety case for UAS operations in support of natural disaster response | Integration, and Operations Conference |
| P62 | Wassyng, A et al. [192] | 2015 | Can Product-Specific Assurance Case Templates Be Used as Medical Device Standards? | IEEE Design & Test |
| P63 | Carlan, C et al. [193] | 2020 | Checkable Safety Cases: Enabling Automated Consistency Checks between Safety Work Products | ISSREW |
| P64 | Hirata, C et al. [194] | 2019 | Combining GSN and STPA for Safety Arguments | ASSURE |
| P65 | Denney, E et al. [195] | 2016 | Composition of safety argument patterns | SAFECOMP |
| P66 | Yuan, T et al. [196] | 2015 | Computer-assisted safety argument review - A dialectics approach | Argument and Computation |
| P67 | Burton, S et al. [197] | 2019 | Confidence Arguments for Evidence of Performance in Machine Learning for Highly Automated Driving Functions | WAISE |
| P68 | Wang, R et al. [198] | 2017 | Confidence assessment framework for safety arguments | SAFECOMP |
| P69 | Groza, A et al. [199] | 2014 | Consistency checking of safety arguments in the Goal Structuring Notation standard | ICCP |
| P70 | Nevsić, D et al. [200] | 2019 | Constructing product-line safety cases from contract-based specifications | SAC |
| P71 | Ray, A et al. [201] | 2013 | Constructing safety assurance cases for medical devices | ASSURE |
| P72 | Warg, F et al. [202] | 2019 | Continuous deployment for dependable systems with continuous assurance cases | ISSREW |
| P73 | Juarez Dominguez, A et al. [203] | 2013 | Creating safety assurance cases for rebreather systems | ASSURE |
| P74 | Chowdhury, T et al. [204] | 2019 | Criteria to Systematically Evaluate (Safety) Assurance Cases | ISSRE |
| P75 | Beyene, T et al. [205] | 2021 | CyberGSN: A Semi-formal Language for Specifying Safety Cases | DSN-W |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P76 | Jaradat, O et al. [206] | 2016 | Deriving Hierarchical Safety Contracts | PRDC |
| P77 | Gallina, B et al. [207] | 2015 | Deriving reusable process-based arguments from process models in the context of railway safety standards | ADA |
| P78 | Gallina, B et al. [208] | 2016 | Deriving safety case fragments for assessing MBASafe's compliance with EN 50128 | SPICE |
| P79 | Sljivo, I et al. [209] | 2015 | Deriving Safety Contracts to Support Architecture Design of Safety Critical Systems | HASE |
| P80 | Gallina, B et al. [210] | 2016 | Deriving verification-related means of compliance for a model-based testing process | DASC |
| P81 | Jia, Y et al. [211] | 2019 | Developing a safety case for electronic prescribing | MEDINFO |
| P82 | Carr, A et al. [212] | 2017 | Developing the Safety Case for MediPi: An Open-Source Platform for Self Management | IHCCLWPH |
| P83 | Luo, Y et al. [213] | 2016 | Development of a safety case editor with assessment features | WASA |
| P84 | Clothier, R et al. [106] | 2015 | Development of a Template Safety Case for Unmanned Aircraft Operations over Populous Areas | SAE Technical papers |
| P85 | Wang, R et al. [214] | 2016 | D-S Theory for argument confidence assessment | BELIEF |
| P86 | Muram, F et al. [122] | 2020 | Dynamic Reconfiguration of Safety-Critical Production Systems | PRDC |
| P87 | Denney, E et al. [120] | 2015 | Dynamic Safety Cases for Through-Life Safety Assurance | ICSE |
| P88 | Diemert, S et al. [215] | 2020 | Eliminative argumentation for arguing system safety - A practitioner's experience | SYSCON |
| P89 | Gallina, B et al. [216] | 2014 | Enabling cross-domain reuse of tool qualification certification artefacts | DEVVARTS |
| P90 | Reich, J et al. [217] | 2020 | Engineering of Runtime Safety Monitors for Cyber-Physical Systems with Digital Dependability Identities | SAFECOMP |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P91 | Mumtaz, M et al. [111] | 2019 | ENGINEERING SAFETY CASE ARGUMENTS USING GSN STANDARDS | JNAS |
| P92 | Cârlan, C et al. [218] | 2020 | Enhancing state-of-the-art safety case patterns to support change impact analysis | ESREL and PSAM |
| P93 | Denney, E et al. [219] | 2013 | Evidence arguments for using formal methods in software certification | ISSREW |
| P94 | Cârlan, C et al. [220] | 2017 | ExplicitCase: Integrated model-based development of system and safety cases | ASSURE |
| P95 | Cârlan, C et al. [221] | 2019 | ExplicitCase: Tool-Support for Creating and Maintaining Assurance Arguments Integrated with System Models | ISSREW |
| P96 | Prokhorova, Y et al. [222] | 2015 | Facilitating construction of safety cases from formal models in Event-B | IST |
| P97 | Jaradat, O et al. [223] | 2016 | Facilitating the Maintenance of Safety Cases | ICRESH-ARMS |
| P98 | Cârlan, C et al. | 2020 [224] | FASTEN.Safe: A Model-Driven Engineering Tool to Experiment with Checkable Assurance Cases. | SAFECOMP |
| P99 | Denney, E et al. [225] | 2015 | Formal Foundations for Hierarchical Safety Cases | HASE |
| P100 | Iliasov, A et al. [226] | 2022 | Formal verification of railway interlocking and its safety case | SCS |
| P101 | Laibinis, L et al. [227] | 2015 | From requirements engineering to safety assurance: Refinement approach | SETTA |
| P102 | Woodham, K et al. [228] | 2018 | FUELEAP model-based system safety analysis | Integration, and Operations Conference |
| P103 | Zeng, F et al. [229] | 2013 | General development framework and its application method for software safety case | Journal of Software |
| P104 | Annable, N et al. [230] | 2022 | Generating Assurance Cases Using Workflow+ Models | SAFECOMP |
| P105 | Sljivo, I et al. [231] | 2014 | Generation of safety case argument-fragments from safety contracts | SAFECOMP |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P106 | Zapata, D et al. [232] | 2018 | Geohazard management approach within safety case | IPC |
| P107 | Chelouati, M et al. [233] | 2022 | Graphical safety assurance case using Goal Structuring Notation (GSN)–challenges, opportunities and a framework for autonomous trains | RESS |
| P108 | Nicolas, C et al. [234] | 2017 | GSN support of mixed-criticality systems certification | DECSoS |
| P109 | Denney, E et al. [235] | 2012 | Heterogeneous aviation safety cases: Integrating the formal and the non-formal | ICECCS |
| P110 | Denney, E et al. [236] | 2013 | Hierarchical safety cases | NFM |
| P111 | Murphy, K et al. [109] | 2012 | How reliable is my safety case? | HAZARDS |
| P112 | Hoang, Q et al. [237] | 2012 | Human-robot interactions: Model-based risk analysis and safety case construction | ERTS |
| P113 | Dardar, R et al. [238] | 2012 | Industrial experiences of building a safety case in compliance with ISO 26262 | ISSREW |
| P114 | Cârlan, C et al. [239] | 2016 | Integrated Formal Methods for Constructing Assurance Cases | ISSREW |
| P115 | Vierhauser, M et al. [103] | 2021 | Interlocking Safety Cases for Unmanned Autonomous Systems in Shared Airspaces | TSE |
| P116 | Lukasiewicz, K et al. [240] | 2018 | Introducing agile practices into development processes of safety critical software | ASD |
| P117 | Despotou, G et al. [241] | 2012 | Introducing safety cases for health IT | SEHC |
| P118 | Ibarra, I et al. [242] | 2012 | ISO 26262 concept phase safety argument for a complex item | SSCS |
| P119 | Kakimoto, K et al. [243] | 2017 | IV&V Case: Empirical study of software independent verification and validation based on safety case | ISSREW |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P120 | Brain, J. M [244] | 2014 | Learning from experience – how can we produce a nuclear safety case to outlast the station? | SSCS |
| P121 | Agrawal, A et al. [104] | 2019 | Leveraging Artifact Trees to Evolve and Reuse Safety Cases | ICSE |
| P122 | Prokhorova, Y et al. [245] | 2012 | Linking modelling in event-B with safety cases | SERENE |
| P123 | Taguchi, K et al. [246] | 2014 | Linking traceability with GSN | ISSREW |
| P124 | Carlan, C [247] | 2017 | Living safety arguments for open systems | ISSREW |
| P125 | Sorokos, I et al. [248] | 2016 | Maintaining Safety Arguments via Automatic Allocation of Safety Requirements | IFAC |
| P126 | Clothier, R et al. [105] | 2017 | Making a risk informed safety case for small unmanned aircraft system operations | ATIO |
| P127 | Nevalainen, R et al. [249] | 2013 | Making Software Safety Assessable and Transparent | EuroSPI |
| P128 | Lin, C et al. [250] | 2018 | Measure Confidence of Assurance Cases in Safety-Critical Domains | ICSE-NIER |
| P129 | Boring, R et al. [251] | 2017 | Measurement sufficiency versus completeness: Integrating safety cases into verification and validation in nuclear control room modernization | AHFE |
| P130 | Jones, P et al. [250] | 2015 | Medical device risk management and safety cases | BMIT |
| P131 | Di Sandro, A et al. [252] | 2020 | MMINT-A 2.0: Tool Support for the Lifecycle of Model-Based Safety Artifacts | MODELS-C |
| P132 | Hartsell, C et al. [253] | 2019 | Model-based design for CPS with learning-enabled components | DESTION |
| P133 | Chouchani, N et al. [4] | 2022 | Model-based safety engineering for autonomous train map | JSS |
| P134 | Retouniotis, A et al. [11] | 2017 | Model-connected safety cases | IMBSA |
| P135 | Denney, E et al. [254] | 2017 | Model-Driven Development of Safety Architectures | MODELS |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P136 | Wang, R et al. [255] | 2018 | Modelling confidence in railway safety case | Safety Science |
| P137 | Larrucea, A et al. [256] | 2017 | Modular Development and Certification of Dependable Mixed-Criticality Systems | DSD |
| P138 | Nevsić, D et al. [257] | 2019 | Modular Safety Cases for Product Lines Based on Assume-Guarantee Contracts | ASSURE |
| P139 | Agarwal, H et al. [258] | 2021 | On safety assurance case for deep learning based image classification in highly automated driving | DATE |
| P140 | Gauerhof, L et al. [259] | 2021 | On the Necessity of Explicit Artifact Links in Safety Assurance Cases for Machine Learning | ISSREW |
| P141 | Denney, E et al. [260] | 2012 | Perspectives on software safety case development for unmanned aircraft | DSN |
| P142 | Gallina, B et al. [261] | 2017 | Pioneering the creation of ISO 26262-compliant OSLC-based safety cases | ISSREW |
| P143 | Katta, V et al. [262] | 2014 | Presenting a traceability based approach for safety argumentation | ESREL |
| P144 | Napolano, M et al. [263] | 2016 | Preventing recurrence of industrial control system accident using assurance case | ISSREW |
| P145 | Chowdhury, T et al. [264] | 2017 | Principles for systematic development of an assurance case template from ISO 26262 | ISSREW |
| P146 | Gallina, B et al. [265] | 2017 | Promoting MBA in the rail sector by deriving process-related evidence via MDSafeCer | CSI |
| P147 | Idmessaoud, Y et al. [266] | 2021 | Quantifying Confidence of Safety Cases with Belief Functions | BELIEF |
| P148 | Nair, S et al. | [267] 2014 | Quantifying uncertainty in safety cases using evidential reasoning | SASSUR |
| P149 | Di Sandro, A et al. [268] | 2019 | Querying automotive system models and safety artifacts with MMINT and viatra | MODELS-C |
| P150 | Denney, E et al. [269] | 2014 | Querying safety cases | SAFECOMP |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P151 | Graydon, P et al. [270] | 2014 | Realistic safety cases for the timing of systems | Computer Journal |
| P152 | Duan, L et al. [271] | 2016 | Representation of Confidence in Assurance Cases Using the Beta Distribution | HASE |
| P153 | Lutz, R. R [272] | 2022 | Requirements Engineering for Safety-Critical Molecular Programs | RE |
| P154 | Sun, L et al. [273] | 2014 | Rethinking of strategy for safety argument development | SASSUR |
| P155 | Guarro, S et al. [274] | 2017 | Risk informed safety case framework for unmanned aircraft system flight software certification | Aerospace |
| P156 | Feng, D et al.[275] | 2013 | Risk-based requirements management framework with applications to assurance cases | Aerospace |
| P157 | Koopman, P et al. [276] | 2019 | Safety argument considerations for public road testing of autonomous vehicles | WCX |
| P158 | Wang, S et al. | [277] 2020 | Safety Argument Pattern Language of Safety-Critical Software | DSA |
| P159 | Fujino, H et al. [278] | 2019 | Safety Assurance Case Description Method for Systems Incorporating Off-Operational Machine Learning and Safety Device | INCOSE |
| P160 | Burton, S et al. [279] | 2021 | Safety Assurance of Machine Learning for Chassis Control Functions | SAFECOMP |
| P161 | Wang, R et al. [280] | 2019 | Safety case confidence propagation based on Dempster–Shafer theory | IJAR |
| P162 | Buysse, L et al. [98] | 2022 | Safety Case Conversion from Goal Structuring Notation to Structured Assurance Case Metamodel | ET |
| P163 | Fahreza Inzaghi, M et al. [281] | 2021 | Safety Case Development for Risk Management of Offshore Pipeline | Earth and Environmental Science |
| P164 | Standish, M et al. [282] | 2014 | Safety case development: a process to implement the safety three-layered framework | SSCS |
| P165 | Ruiz, A et al. [283] | 2015 | Safety case driven development for medical devices | SAFECOMP |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P166 | Holmberg, J et al. [284] | 2012 | Safety case framework to provide justifiable reliability numbers for software systems | ESREL and PSAM |
| P167 | Kokaly, S et al. [285] | 2017 | Safety case impact assessment in automotive software systems: An improved model-based approach | SAFECOMP |
| P168 | Ilizástigui Pérez, F [286] | 2020 | Safety case process in Cuba: Transition from theory to practice | PSEP |
| P169 | Ilizástigui Pérez, F [110] | 2017 | Safety Case regulations for major hazard facilities in Cuba | HAZARDS |
| P170 | Birch, J et al. [114] | 2013 | Safety cases and their role in ISO 26262 functional safety assessment | SAFECOMP |
| P171 | Mirzaei, E et al. [287] | 2020 | Safety cases for adaptive systems of systems: State of the art and current challenges | EDCC |
| P172 | Sujan, M et al. [288] | 2013 | Safety cases for medical devices and health information technology: Involving health-care organisations in the assurance of safety | Health Informatics |
| P173 | Denney, E et al. [289] | 2016 | Safety considerations for UAS ground-based detect and avoid | DASC |
| P174 | Nair, S et al. [290] | 2014 | Safety evidence traceability: Problem analysis and model | REFSQ |
| P175 | Heikkilä, E et al. [291] | 2017 | Safety qualification process for an autonomous ship prototype - A goal-based safety case approach | TRANSNAV |
| P176 | Ratiu, D et al. [292] | 2015 | Safety.Lab: Model-based domain specific tooling for safety argumentation. | ASSURE |
| P177 | Meyer, M et al. [293] | 2022 | Scenario- and Model-Based Systems Engineering Procedure for the SOTIF-Compliant Design of Automated Driving Functions | IV |
| P178 | Aiello, M et al. [294] | 2014 | SCT: A safety case toolkit | ISSREW |
| P179 | Socha, K et al. [2] | 2022 | SMIRK: A machine learning-based pedestrian automatic emergency braking system with a complete safety case | Software Impacts |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P180 | Zeng, F et al. [295] | 2012 | Software safety certification framework based on safety case | CSSS |
| P181 | Schwalbe, G et al. [296] | 2020 | Structuring the Safety Argumentation for Deep Neural Network Based Perception in Automotive Applications | WAISE |
| P182 | Lin, C et al. [297] | 2017 | Support for safety case generation via model transformation | SIGBED |
| P183 | Górski, J et al. [298] | 2012 | Supporting assurance by evidence-based argument services | ERCIM |
| P184 | de Oliveira, A et al. [299] | 2015 | Supporting the automated generation of modular product line safety cases | DepCoS-RELCOMEX |
| P185 | Bagheri, H et al. [300] | 2020 | Synthesis of Assurance Cases for Software Certification | ICSE NIER |
| P186 | Chowdhury, T et al. [301] | 2020 | Systematic Evaluation of (Safety) Assurance Cases | SAFECOMP |
| P187 | Jaradat, O et al. [302] | 2016 | Systematic maintenance of safety cases to reduce risk | ASSURE |
| P188 | Matsuno, Y et al. [303] | 2019 | Tackling Uncertainty in Safety Assurance for Machine Learning: Continuous Argument Engineering with Attributed Tests | WAISE |
| P189 | Stalhane, T et al. [304] | 2016 | The agile safety case | ASSURE |
| P190 | Cassano, V et al. [305] | 2014 | The definition and assessment of a safety argument | ISSREW |
| P191 | Yang, J et al. [306] | 2017 | The Development of Safety Cases for an Autonomous Vehicle: A Comparative Study on Different Methods | ICVS |
| P192 | Sujan, M et al. [307] | 2015 | The development of safety cases for healthcare services: Practical experiences, opportunities and challenges | RESS |
| P193 | Ellor, G [308] | 2021 | The e-SafetyCase - Electronic or Effortless? | CCPS 2021 |
| P194 | Viger, T et al. [309] | 2022 | The ForeMoSt approach to building valid model-based safety arguments | SoSym |
| P195 | Salay, R et al. [310] | 2022 | The Missing Link: Developing a Safety Case for Perception Components in Automated Driving | WCX |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P196 | Denney, E et al. [311] | 2019 | The role of safety architectures in aviation safety cases | RESS |
| P197 | Standish, M et al. [312] | 2014 | The safety three-layer framework: a case study | SSCS |
| P198 | Denney, E et al. [113] | 2018 | Tool support for assurance case development | ASE |
| P199 | Ruiz, A et al. [**?**] | 2012 | Towards a case-based reasoning approach for safety assurance reuse | SASSUR |
| P200 | Graydon, P. J [313] | 2014 | Towards a clearer understanding of context and its role in assurance argument confidence | SAFECOMP |
| P201 | Denney, E et al. [314] | 2015 | Towards a formal basis for modular safety cases | SAFECOMP |
| P202 | McDermid, J et al. [315] | 2019 | Towards a framework for safety assurance of autonomous systems | CEUR |
| P203 | Geissler, F et al. [316] | 2021 | Towards a safety case for hardware fault tolerance in convolutional neural networks using activation range supervision | CEUR |
| P204 | Eastwood, R et al. [317] | 2013 | Towards a safety case for runtime risk and uncertainty management in safety-critical systems | CSC |
| P205 | Gallina, B et al. [318] | 2017 | Towards an ISO 26262-compliant OSLC-based tool chain enabling continuous self-assessment | QUATIC |
| P206 | Brito, M. P [319] | 2017 | Towards building a safety case for marine unmanned surface vehicles: A bayesian perspective | ESREL |
| P207 | Shahin, R et al. [320] | 2021 | Towards Certified Analysis of Software Product Line Safety Cases | SAFECOMP |
| P208 | Alajrami, S et al. [321] | 2016 | Towards cloud-based enactment of safety-related processes | SAFECOMP |
| P209 | Javed, M et al. [100] | 2021 | Towards dynamic safety assurance for Industry 4.0 | JSA |
| P210 | Gallina, B et al. [322] | 2015 | Towards Enabling Reuse in the Context of Safety-Critical Product Lines | PLEASE |
| P211 | Wardziński, A et al. [323] | 2016 | Towards safety case integration with hazard analysis for medical devices | ASSURE |

| No | Author(s) | Year | Title | Venue |
|---|---|---|---|---|
| P212 | Idmessaoud, Y et al. [324] | 2022 | Uncertainty Elicitation and Propagation in GSN Models of Assurance Cases | SAFECOMP |
| P213 | Mjeda, A et al. [325] | 2020 | Uncertainty Entangled; Modelling Safety Assurance Cases for Autonomous Systems | ETAPS |
| P214 | Wardziński, A et al. [326] | 2017 | Uniform model interface for assurance case integration with system models | ASSURE |
| P215 | Pissoort, D et al. [327] | 2019 | Use of the Goal Structuring Notation (GSN) as Generic Notation for an 'EMC Assurance Case' | EMC |
| P216 | Kläs, M et al. [328] | 2021 | Using complementary risk acceptance criteria to structure assurance cases for safety-critical AI components | CEUR |
| P217 | Zeng, F et al. [329] | 2013 | Using D-S evidence theory to evaluation of confidence in safety case | JTAIT |
| P218 | Jaradat, O et al. [330] | 2017 | Using Safety Contracts to Guide the Maintenance of Systems and Safety Cases | EDCC |
| P219 | Jaradat, O et al. [331] | 2018 | Using safety contracts to verify design assumptions during runtime | ICRST |
| P220 | Jaradat, O et al. [332] | 2015 | Using sensitivity analysis to facilitate the maintenance of safety cases | ICRST |
| P221 | Ferrell, U et al. [123] | 2022 | Validation of Assurance Case for Dynamic Systems | DASC |
| P222 | Brain, J [333] | 2012 | Visual representation of safety cases | Measurement and Control |
| P223 | Bragg, J et al. [334] | 2018 | What is acceptably safe for reinforcement learning | WAISE |
| P224 | X. Zhao et al. [335] | 2012 | A new approach to assessment of confidence in assurance cases | SASSUR |

## B Abbreviations of the conferences and journals

- SAFECOMP: International Conference on Computer Safety, Reliability, and Security
- ISSREW: International Symposium on Software Reliability Engineering Workshop
- ICSE: International Conference on Software Engineering
- HASE: High-Assurance Systems Engineering
- EDCC: European Dependable Computing Conference
- ASSURE : International Workshop on Assurance Cases for Software-Intensive Systems
- SSCS : International System Safety Conference, incorporating the Cyber-Security Conference
- HAZARDS : Institution of Chemical Engineers Symposium on Hazards
- JSA : Journal of Systems Architecture
- BELIEF : International Conference on Belief Functions
- ESREL : European Safety and Reliability Conference
- PSAM : Probabilistic Safety Assessment and Management Conference
- ATIO : AIAA Aviation Technology, Integration, and Operations Conference
- NFM : International Symposium on NASA Formal Methods
- KES : International Conference on Knowledge-Based and Intelligent Information & Engineering Systems
- Ada-Europe : Ada-Europe International Conference on Reliable Software Technologies
- SEH : International Workshop on Software Engineering in Health Care
- DepCoS-RELCOMEX : International Conference on Dependability and Complex Systems
- Journal of Software
- DSA : International Conference on Dependable Systems and Their Applications
- REFSQ : International Working Conference on Requirements Engineering: Foundation for Software Quality
- DSD : Euromicro Conference on Digital System Design
- WAISE : International Workshop on Artificial Intelligence Safety Engineering
- DSN : IEEE/IFIP International Conference on Dependable Systems and Networks

- AEROTECH : SAE AeroTech Congress and Exhibition
- DSN-W : IEEE/IFIP International Conference on Dependable Systems and Networks Workshops
- ASE : Automated Software Engineering
- DATE : Design, Automation & Test in Europe Conference & Exhibition
- SAC : ACM/SIGAPP Symposium on Applied Computing
- BI&T : Biomedical Instrumentation and Technology
- CSSS : International Conference on Computer Science and Service System
- ET : International Scientific Conference Electronics
- WCX: Annual World Congress Experience
- EuroPLoP: European Conference on Pattern Languages of Programs
- ISSRE : International Symposium on Software Reliability Engineering
- AISafety : Workshop on Artificial Intelligence Safety
- JATIT : Journal of Theoretical and Applied Information Technology
- CSI : Computer Standard Interfaces
- DASC : Digital Avionics Systems Conference
- MODELS : International Conference on Model Driven Engineering Languages and Systems
- JSS : Journal of Systems and Software
- PLEASE : International Workshop on Product Line Approaches in Software Engineering
- MEDINFO : World Congress on Medical and Health Informatics
- PSEP : Process Safety and Environmental Protection
- CIT : International Conference on Computer and Information Technology
- PRDC : IEEE Pacific Rim International Symposium on Dependable Computing
- ECCS : International Conference on Engineering of Complex Computer Systems
- ICFEM: International Conference on Formal Engineering Methods
- RE : International Requirements Engineering Conference
- SIGBED : ACM Special Interest Group on Embedded Systems

- RESS : Reliability Engineering & System Safety
- AHFE : Advances in Human Factors in Energy: Oil, Gas, Nuclear and Electric Power Industries, Advances in Intelligent Systems and Computing
- AIAA : American Institute of Aeronautics and Astronautics
- SERENE : International conference on Software Engineering for Resilient Systems
- TRE : IEEE Transactions on Reliability
- SPICE : International Conference on Software Process Improvement and Capability Determination
- IJAR : International Journal of Approximate Reasoning
- IMBSA : Model-Based Automatic Generation of Safety Arguments
- TRANSNAV : International Conference on Marine Navigation and Safety of Sea Transportation: Marine Navigation
- INCOSE : International Council on Systems Engineering
- TSE : IEEE Transactions on Software Engineering
- WASA : Workshop on Automotive Systems/Software Architectures
- ESSAT : European Association of Software Science and Technology
- SETTA : International Symposium on Dependable Software Engineering: Theories, Tools, and Applications
- IPC : International Symposium on Dependable Software Engineering: Theories, Tools, and Applications
- ISOCEEN : International Seminar on Ocean and Coastal Engineering, Environmental and Natural Disaster Management
- ICRST : International Conference on Reliable Software Technologies
- JTAIT : Journal of Theoretical & Applied Information Technology
- CEUR Workshops
- QUATIC : International Conference on the Quality of Information and Communications Technology
- CSC: International System Safety Conference incorporating the Cyber Security Conference
- SoSym : Software and Systems Modelling Journal
- CCPS : Center for Chemical Process Safety International Conference

- ICVS : Intelligent and Connected Vehicles Symposium
- ERCIM : European Research Consortium for Informatics and Mathematics
- IV : IEEE Symposium on Intelligent Vehicle
- DSA : International Conference on Dependable Systems and Their Applications
- SASSUR : International Workshop on Next Generation of System Assurance Approaches for Critical Systems
- DESTION : Workshop on Design Automation for CPS and IoT
- BMIT : Biomedical Instrumentation & Technology
- AHFE : Advances in Human Factors in Energy: Oil, Gas, Nuclear and Electric Power Industries
- EuroSPI : European Conference on System and Software Process Improvement
- IFAC : International Federation of Automatic Control
- SEHC : International Workshop on Software Engineering in Health Care
- ASD : International Conference on Agile Software Development
- ERTS : Embedded Real Time Software and Systems
- SCS : Safety-Critical Systems
- ICRESH-ARMS : Current Trends in Reliability, Availability, Maintainability and Safety
- IST : Information and Software Technology
- JNAS : Journal of Natural and Applied Sciences
- DEVVARTS : Development, Verification and Validation of Critical Systems
- SYSCON : IEEE International Systems Conference
- IHCCLWPH : Informatics for Health: Connected Citizen-Led Wellness and Population Health
- SPICE : Software Process Improvement and Capability Determination
- SUM : Scalable Uncertainty Management
- IJCCBS : International Journal of Critical Computer-Based Systems
- ICIT : Internation Conference on Industrial Technology
- PHM : Prognostics and System Health Management Conference
- ICSRS : International Conference on System Reliability and Safety
- MCPS : Workshop on Medical Cyber Physical Systems

# Bibliography


[1] M. Yuan, J. Shan, and K. Mi, "Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 818–825, 2021.

[2] K. Socha, M. Borg, and J. Henriksson, "Smirk: A machine learning-based pedestrian automatic emergency braking system with a complete safety case," *Software Impacts*, vol. 13, p. 100352, 2022.

[3] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli, "Guidance on the assurance of machine learning in autonomous systems (amlas),"

[4] N. Chouchani, S. Debbech, and M. Perin, "Model-based safety engineering for autonomous train map," *Journal of Systems and Software*, vol. 183, 2022.

[5] I. I. organization for standardization, "Iso 26262-3:2018 road vehicles, functional safety, part-3: Concept phase," 2018.

[6] L. Duan, S. Rayadurgam, M. Heimdahl, O. Sokolsky, and I. Lee, "Representation of confidence in assurance cases using the beta distribution," in *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 86–93, IEEE, 2016.

[7] S. Burton, I. Kurzidem, A. Schwaiger, P. Schleiss, M. Unterreiner, T. Graeber, and P. Becker, "Safety assurance of machine learning for chassis control functions," in *Computer Safety, Reliability, and Security: 40th International Conference, SAFECOMP 2021, York, UK, September 8–10, 2021, Proceedings 40*, pp. 149–162, Springer, 2021.

[8] A. Di Sandro, S. Kokaly, R. Salay, and M. Chechik, "Querying automotive system models and safety artifacts with mmint and viatra," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 2–11, IEEE, 2019.

[9] P. Koopman, "Autonomous vehicles and software safety engineering," in *ICSE keynote, May 2022*, 2022.

[10] M. Pratt, "Titan submarine disaster," 2023. `https://www.pbs.org/newshour/nation/experts-say-the-titan-subs-unconventional-design-may-have-destined-it-for-disaster`.

[11] A. Retouniotis, Y. Papadopoulos, I. Sorokos, D. Parker, N. Matragkas, and S. Sharvia, "Model-connected safety cases," in *Model-Based Safety and Assessment: 5th International Symposium, IMBSA 2017, Trento, Italy, September 11–13, 2017, Proceedings 5*, vol. 10437 LNCS, pp. 50–63, 2017.

[12] M. Maksimov, S. Kokaly, and M. Chechik, "A survey of tool-supported assurance case assessment techniques," *ACM Computing Surveys*, vol. 52, no. 5, 2019.

[13] Y. Matsuno, T. Takai, and S. Yamamoto, "Facilitating use of assurance cases in industries by workshops with an agent-based method," *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 6, pp. 1297–1308, 2020.

[14] R. Palin, D. Ward, I. Habli, and R. Rivett, "Iso 26262 safety cases: Compliance and assurance," in *6th IET International Conference on System Safety 2011*, 2011.

[15] R. Bell, "Introduction to iec 61508," in *Acm international conference proceeding series*, vol. 162, pp. 3–12, Citeseer, 2006.

[16] J. L. de La Vara, M. Borg, K. Wnuk, and L. Moonen, "An industrial survey of safety evidence change impact analysis practice," *IEEE TSE*, vol. 42, no. 12, pp. 1095–1117, 2016.

[17] J. Górski, A. Jarzkebowicz, R. Leszczyna, J. Miler, and M. Olszewski, "Trust case: Justifying trust in an it solution," *Reliability Engineering & System Safety*, vol. 89, no. 1, pp. 33–47, 2005.

[18] M. Chelouati, A. Boussif, J. Beugin, and E.-M. El Koursi, "Graphical safety assurance case using goal structuring notation (gsn)—challenges, opportunities and a framework for autonomous trains," *Reliability Engineering & System Safety*, vol. 230, p. 108933, 2023.

[19] Y. Qin, Z. Xu, X. Wang, and M. vSkare, "Green energy adoption and its determinants: A bibliometric analysis," *Renewable and Sustainable Energy Reviews*, vol. 153, p. 111780, 2022.

[20] M. K. Linnenluecke, M. Marrone, and A. K. Singh, "Conducting systematic literature reviews and bibliometric analyses," *Australian Journal of Management*, vol. 45, no. 2, pp. 175–194, 2020.

[21] A. V. S. Neto, J. B. Camargo, J. R. Almeida, and P. S. Cugnasca, "Safety assurance of artificial intelligence-based systems: A systematic literature review on the state of the art and guidelines for future work," *IEEE Access*, vol. 10, pp. 130733–130770, 2022.

[22] W. E. Wong, N. Mittas, E. M. Arvanitou, and Y. Li, "A bibliometric assessment of software engineering themes, scholars and institutions (2013–2020)," *Journal of Systems and Software*, vol. 180, p. 111029, 2021.

[23] F. Deng, J. Jiang, and I. Sirés, "State-of-the-art review and bibliometric analysis on electro-fenton process," *Carbon Letters*, vol. 33, no. 1, pp. 17–34, 2023.

[24] E. Denney, G. Pai, and J. Pohl, "AdvoCATE: An assurance case automation toolset," *Computer Safety, Reliability, and Security: SAFECOMP 2012 Workshops: Sassur, ASCoMS, DESEC4LCCI, ERCIM/EWICS, IWDE, Magdeburg, Germany, September 25-28, 2012. Proceedings 31*, pp. 8–21, 2012.

[25] M. Maksimov, S. Kokaly, and M. Chechik, "A survey of tool-supported assurance case assessment techniques," *ACM Computing Surveys*, vol. 52, no. 5, 2019.

[26] C. Menghi, T. Viger, A. Di Sandro, C. Rees, J. Joyce, and M. Chechik, "Assurance case development as data: A manifesto," in *2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 135–139, IEEE, 2023.

[27] S. Ramakrishna, H. Jin, A. Dubey, and A. Ramamurthy, "Automating pattern selection for assurance case development for cyber-physical systems," in *International Conference on Computer Safety, Reliability, and Security*, pp. 82–96, Springer, 2022.

[28] S. Nair, J. L. De la Vara, M. Sabetzadeh, and L. Briand, "Classification, structuring, and assessment of evidence for safety–a systematic literature review," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, pp. 94–103, IEEE, 2013.

[29] B. Chen, K. Chen, S. Hassani, Y. Yang, D. Amyot, L. Lessard, G. Mussbacher, M. Sabetzadeh, and D. Varró, "On the use of gpt-4 for creating goal models: an exploratory study," in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, pp. 262–271, IEEE, 2023.

[30] OpenAI, "Gpt 4," 2023.

[31] B. Chen, K. Chen, S. Hassani, Y. Yang, D. Amyot, L. Lessard, G. Mussbacher, M. Sabetzadeh, and D. Varró, "On the use of gpt-4 for creating goal models: an exploratory study," in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, pp. 262–271, IEEE, 2023.

[32] OMG, "Structured assurance case metamodel (version 2.2)," 2021.

[33] N. Mansourov and D. Campara, *System assurance: beyond detecting vulnerabilities*. Elsevier, 2010.

[34] T. Chowdhury, A. Wassyng, R. F. Paige, and M. Lawford, "Criteria to systematically evaluate (safety) assurance cases," in *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, vol. 2019-October, pp. 380–390, 2019.

[35] S. Foster, Y. Nemouchi, M. Gleirscher, R. Wei, and T. Kelly, "Integration of formal proof into unified assurance cases with isabelle/sacm," *Formal Aspects of Computing*, vol. 33, no. 6, pp. 855–884, 2021.

[36] A. B. Belle, T. C. Lethbridge, S. Kpodjedo, O. O. Adesina, and M. A. Garzon, "A novel approach to measure confidence and uncertainty in assurance cases," *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pp. 24–33, 2019.

[37] Y. Matsuno, T. Takai, and S. Yamamoto, "Facilitating use of assurance cases in industries by workshops with an agent-based method," *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 6, pp. 1297–1308, 2020.

[38] M. Mohamad, J.-P. Steghöfer, and R. Scandariato, "Security assurance cases—state of the art of an emerging approach," *EMSE*, vol. 26, no. 4, p. 43, 2021.

[39] R. Alexander, R. Hawkins, and T. Kelly, "Security assurance cases: motivation and the state of the art," *High Integrity Systems Engineering Department of Computer Science University of York Deramore Lane York YO10 5GH*, 2011.

[40] M. Zeroual, B. Hamid, M. Adedjouma, and J. Jaskolka, "Constructing security cases based on formal verification of security requirements in alloy," in *International Conference on Computer Safety, Reliability, and Security*, pp. 15–25, Springer, 2023.

[41] A. Ray and R. Cleaveland, "Constructing safety assurance cases for medical devices," in *ASSURE*, pp. 40–45, IEEE, 2013.

[42] C. Menghi, T. Viger, A. Di Sandro, C. Rees, J. Joyce, and M. Chechik, "Assurance case development as data: A manifesto," in *2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 135–139, IEEE, 2023.

[43] S. Ramakrishna, H. Jin, A. Dubey, and A. Ramamurthy, "Automating pattern selection for assurance case development for cyber-physical systems," in *International Conference on Computer Safety, Reliability, and Security*, pp. 82–96, Springer, 2022.

[44] S. Nair, J. L. de la Vara, M. Sabetzadeh, and L. Briand, "Classification, structuring, and assessment of evidence for safety–a systematic literature review," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, pp. 94–103, IEEE, 2013.

[45] T. Stralhane and T. Myklebust, "The agile safety case," in *Computer Safety, Reliability, and Security: SAFECOMP 2016 Workshops, ASSURE, DECSoS, SASSUR, and TIPS, Trondheim, Norway, September 20, 2016, Proceedings 35*, pp. 5–16, Springer, 2016.

[46] C. M. Holloway, "Safety case notations: Alternatives for the non-graphically inclined?," in *2008 3rd IET International Conference on System Safety*, pp. 1–6, IET, 2008.

[47] C. Carlan and B. Gallina, "Enhancing state-of-the-art safety case patterns to support change impact analysis," in *30th European Safety and Reliability Conference*, 2020.

[48] P. Schleiss, F. Carella, and I. Kurzidem, "Towards continuous safety assurance for autonomous systems," in *2022 6th International Conference on System Reliability and Safety (ICSRS)*, pp. 457–462, IEEE, 2022.

[49] ACWG, "Goal structuring notation standard v.3." `https://scsc.uk/r141C:1?t=1`, 2021.

[50] M. Vierhauser, S. Bayley, J. Wyngaard, W. Xiong, J. Cheng, J. Huseman, R. Lutz, and J. Cleland-Huang, "Interlocking safety cases for unmanned autonomous systems in shared airspaces," *IEEE transactions on software engineering*, vol. 47, no. 5, pp. 899–918, 2019.

[51] R. Alexander, R. Hawkins, and T. Kelly, "Security assurance cases: motivation and the state of the art," *High Integrity Systems Engineering Department of Computer Science University of York Deramore Lane York YO10 5GH*, 2011.

[52] M. Sivakumar, A. B. Belle, J. Shan, and K. K. Shahandashthi, "Exploring the use of gpt-4 to automatically generate safety cases : a preliminary study," *submitted to FORGE 2024*, 2024.

[53] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[54] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of bert in ranking," *arXiv preprint arXiv:1904.07531*, 2019.

[55] S. Nair, J. L. De La Vara, M. Sabetzadeh, and L. Briand, "An extended systematic literature review on provision of evidence for safety certification," *Information and Software Technology*, vol. 56, no. 7, pp. 689–717, 2014.

[56] F. Tambon, G. Laberge, L. An, A. Nikanjam, P. S. N. Mindom, Y. Pequignot, F. Khomh, G. Antoniol, E. Merlo, and F. Laviolette, "How to certify machine learning based safety-critical systems? a systematic literature review," *Automated Software Eng.*, vol. 29, no. 2, p. 74, 2022.

[57] J. Han, H.-J. Kang, M. Kim, and G. H. Kwon, "Mapping the intellectual structure of research on surgery with mixed reality: Bibliometric network analysis (2000–2019)," *Journal of Biomedical Informatics*, vol. 109, p. 103516, 2020.

[58] E. Denney and G. Pai, "Automating the assembly of aviation safety cases," *IEEE Transactions on Reliability*, vol. 63, no. 4, pp. 830–849, 2014.

[59] E. Denney and G. Pai, "Tool support for assurance case development," *Automated Software Engineering*, vol. 25, no. 3, pp. 435–499, 2018.

[60] E. Denney and G. Pai, "Evidence arguments for using formal methods in software certification," in *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 375–380, IEEE, 2013.

[61] C. Hartsell, N. Mahadevan, A. Dubey, and G. Karsai, "Automated method for assurance case construction from system design models," in *2021 5th International Conference on System Reliability and Safety (ICSRS)*, pp. 230–239, IEEE, 2021.

[62] T. E. Wang, C. Oh, M. Low, I. Amundson, Z. Daw, A. Pinto, M. L. Chiodo, G. Wang, S. Hasan, R. Melville, *et al.*, "Computer-aided generation of assurance cases," in *International Conference on Computer Safety, Reliability, and Security*, pp. 135–148, Springer, 2023.

[63] S. Ramakrishna, C. Hartsell, A. Dubey, P. Pal, and G. Karsai, "A methodology for automating assurance case generation," *arXiv preprint arXiv:2003.05388*, 2020.

[64] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the safety of machine learning for pedestrian detection at crossings," in *Computer Safety, Reliability, and Security: 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, September 16–18, 2020, Proceedings 39*, pp. 197–212, Springer, 2020.

[65] R. Ashmore, R. Calinescu, and C. Paterson, "Assuring the machine learning lifecycle: Desiderata, methods, and challenges," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–39, 2021.

[66] M. Weyssow, H. Sahraoui, and E. Syriani, "Recommending metamodel concepts during modeling activities with pre-trained language models," *Software and Systems Modeling*, vol. 21, no. 3, pp. 1071–1089, 2022.

[67] M. B. Chaaben, L. Burgueño, and H. Sahraoui, "Towards using few-shot prompt learning for automating model completion," in *2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 7–12, IEEE, 2023.

[68] J. Cámara, J. Troya, L. Burgueño, and A. Vallecillo, "On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml," *Software and Systems Modeling*, pp. 1–13, 2023.

[69] T. Viger, L. Murphy, S. Diemert, C. Menghi, A. Di, and M. Chechik, "Supporting assurance case development using generative ai," in *SAFECOMP 2023, Position Paper*, 2023.

[70] A. Ahmad, M. Waseem, P. Liang, M. Fahmideh, M. S. Aktar, and T. Mikkonen, "Towards human-bot collaborative software architecting with chatgpt," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, pp. 279–285, 2023.

[71] R. Djaber and H. Ismail, "Ai as a co-engineer: A case study of chatgpt in software lifecycle," 2023.

[72] H.-G. Fill, P. Fettke, and J. Köpke, "Conceptual modeling and large language models: impressions from first experiments with chatgpt," *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, vol. 18, pp. 1–15, 2023.

[73] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews.," *International journal of surgery*, vol. 88, p. 105906, 2021.

[74] B. Kitchenham, "Segress: Software engineering guidelines for reporting secondary studies.," *IEEE Transactions on Software Engineering.*, vol. 49, pp. 1273–1298., 2022.

[75] T. Gotschall, "Endnote 20 desktop version.," *Journal of the Medical Library Association: JMLA*, vol. 109(3), 2020.

[76] IEEE, "Ieee xplore," 2023.

[77] A. of Computing Machinery, "Acm digital library," 2023.

[78] Elsevier, "Scopus," 2023.

[79] Elsevier, "Engineering village (ev)," 2023.

[80] Google, "Google scholar," 2023.

[81] A. Harzing, "Publish or perish software," 2007. figshare http://www.harzing.com/pop.htm.

[82] T. Gotschall, "Endnote 20 desktop version.," *Journal of the Medical Library Association: JMLA*, vol. 109(3), 2020.

[83] M. K. Linnenluecke, M. Marrone, and A. K. Singh, "Conducting systematic literature reviews and bibliometric analyses," *Australian Journal of Management*, vol. 45, no. 2, pp. 175–194, 2020.

[84] A. V. S. Neto, J. B. Camargo, J. R. Almeida, and P. S. Cugnasca, "Safety assurance of artificial intelligence-based systems: A systematic literature review on the state of the art and guidelines for future work," *IEEE Access*, 2022.

[85] B. D. Catumba, M. B. Sales, P. T. Borges, M. N. Ribeiro Filho, A. A. S. Lopes, M. A. de Sousa Rios, A. S. Desai, M. Bilal, and J. C. S. dos Santos, "Sustainability and challenges in hydrogen production: An advanced bibliometric analysis," *International Journal of Hydrogen Energy*, vol. 48, no. 22, pp. 7975–7992, 2023.

[86] F. Deng, J. Jiang, and I. Sirés, "State-of-the-art review and bibliometric analysis on electro-fenton process," *Carbon Letters*, vol. 33, no. 1, pp. 17–34, 2023.

[87] S. Khanra, A. Dhir, and M. Mäntymäki, "Big data analytics and enterprises: a bibliometric synthesis of the literature," *Enterprise Information Systems*, vol. 14, no. 6, pp. 737–768, 2020.

[88] N. Van Eck and L. Waltman, "Software survey: Vosviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010.

[89] G. Charts, "Sankey diagram," 2023.

[90] S. Batt, T. Grealis, O. Harmon, and P. Tomolonis, "Learning tableau: A data visualization tool," *The Journal of Economic Education*, vol. 51, no. 3-4, pp. 317–328, 2020.

[91] Quanser, "Qcar."

[92] K. Beckers, M. Heisel, T. Frese, and D. Hatebur, "A structured and model-based hazard analysis and risk assessment method for automotive systems," in *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 238–247, IEEE, 2013.

[93] R. Debouk, "Overview of the second edition of iso 26262: Functional safety—road vehicles," *Journal of System Safety*, vol. 55, no. 1, pp. 13–21, 2019.

[94] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[95] OpenAI, "Prompt engineering." https://platform.openai.com/docs/guides/prompt-engineering, 2023.

[96] R. B. Nelsen, "Kendall tau metric," *Encyclopedia of Mathematics*, 2001.

[97] R. Wei, T. P. Kelly, X. Dai, S. Zhao, and R. Hawkins, "Model based system assurance using the structured assurance case metamodel," *Journal of Systems and Software*, vol. 154, pp. 211–233, 2019.

[98] L. Buysse, D. Vanoost, J. Vankeirsbilck, J. Boydens, and D. Pissoort, "Safety case conversion from goal structuring notation to structured assurance case metamodel," in *2022 XXXI International Scientific Conference Electronics (ET)*, pp. 1–5, IEEE, 2022.

[99] P. Schleiss, F. Carella, and I. Kurzidem, "Towards continuous safety assurance for autonomous systems," in *ICSRS*, pp. 457–462, IEEE, 2022.

[100] M. A. Javed, F. U. Muram, H. Hansson, S. Punnekkat, and H. Thane, "Towards dynamic safety assurance for industry 4.0," *Journal of Systems Architecture*, vol. 114, p. 101914, 2021.

[101] E. Fox, "Tesla robotaxi could become the company's greatest value creator," 2023.

[102] L. Zhang, "The cruise safety report," 2022.

[103] M. Vierhauser, S. Bayley, J. Wyngaard, W. Xiong, J. Cheng, J. Huseman, R. Lutz, and J. Cleland-Huang, "Interlocking safety cases for unmanned autonomous systems in shared airspaces," *IEEE TSE*, vol. 47(5), pp. 899–918, 2019.

[104] A. Agrawal, S. Khoshmanesh, M. Vierhauser, M. Rahimi, J. Cleland-Huang, and R. Lutz, "Leveraging artifact trees to evolve and reuse safety cases," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 1222–1233, IEEE, 2019.

[105] R. Clothier, E. Denney, and G. J. Pai, "Making a risk informed safety case for small unmanned aircraft system operations," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3275, 2017.

[106] R. Clothier, B. Williams, and A. Washington, "Development of a template safety case for unmanned aircraft operations over populous areas," tech. rep., SAE Technical Paper, 2015.

[107] Amazon, "Amazon prime air prepares for drone deliveries," 2022.

[108] G. Macher, H. Sporer, and C. Kreiner, "Automotive safety case pattern," in *19th European Conference on Pattern Languages of Programs - EuroPLoP '14*, p. Article 12, Association for Computing Machinery, 2014.

[109] K. R. Murphy, S. M. Gilbert, and B. Lewis, "How reliable is my safety case?," in *23rd Institution of Chemical Engineers Symposium on Hazards*, pp. 392–398, 2012.

[110] I. P. Fidel, "Safety case regulations for major hazard facilities in cuba," in *Hazards 27*, vol. 2017-May, Institution of Chemical Engineers, 2017.

[111] M. Maidah, A. Sidra, M. Nahal, and K. Tabassum, "Engineering safety case arguments using gsn standards," *Journal of Natural and Applied Sciences Pakistan*, vol. 1, no. 1, pp. 124–131, 2019.

[112] E. Gareth, "The e-safetycase – electronic or effortless?," in *36th Center for Chemical Process Safety International Conference, CCPS 2021*, Topical Conference at the 2021 AIChE Spring Meeting and 17th Global Congress on Process Safety, 2021.

[113] E. Denney and G. Pai, "Tool support for assurance case development," *Automated Software Engineering*, vol. 25, no. 3, pp. 435–499, 2018.

[114] J. Birch, R. Rivett, I. Habli, B. Bradshaw, J. Botham, D. Higham, P. Jesty, H. Monkhouse, and R. Palin, "Safety cases and their role in iso 26262 functional safety assessment," in *SAFECOMP*, pp. 154–165, Springer, 2013.

[115] B. Brosgol, "Do-178c: the next avionics safety standard," *ACM SIGAda Ada Letters*, vol. 31, no. 3, pp. 5–6, 2011.

[116] T. K. Ferrell and U. D. Ferrell, "Rtca do-178b/eurocae ed-12b," in *Digital Avionics Handbook*, pp. 467–478, CRC Press, 2000.

[117] C. Willis, "Ensuring the safety of nuclear installations: Lessons learned from the fukushima daiichi accident," 2021.

[118] C. digital, "The norwegian natural catastrophe compensation system in the past and the future," 2017.

[119] N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control," in *2018 Annual American Control Conference (ACC)*, pp. 3215–3220, IEEE, 2018.

[120] E. Denney, G. Pai, and I. Habli, "Dynamic safety cases for through-life safety assurance," in *ICSE*, vol. 2, pp. 587–590, 2015.

[121] A. B. Belle, H. Hemmati, and T. C. Lethbridge, "Position paper: a vision for the dynamic safety assurance of ml-enabled autonomous driving systems," 2023.

[122] F. U. Muram, M. A. Javed, H. Hansson, and S. Punnekkat, "Dynamic reconfiguration of safety-critical production systems," in *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC*, pp. 120–129, IEEE Computer Society, 2020.

[123] U. D. Ferrell and A. H. A. Anderegg, "Validation of assurance case for dynamic systems," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pp. 1–11, IEEE, 2022.

[124] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[125] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang, "A map of threats to validity of systematic literature reviews in software engineering," in *APSEC*, pp. 153–160, IEEE, 2016.

[126] C. Kang, "The new york times company: Openai's ceo urges regulation for artificial intelligence," 2023.

[127] P. Koopman, "Ul 4600: What to include in an autonomous vehicle safety case," *Computer*, vol. 56, no. 05, pp. 101–104, 2023.

[128] Y. Wang, I. Bogicevic, and S. Wagner, "A study of safety documentation in a scrum development process," in *Proceedings of the XP2017 Scientific Workshops*, pp. 1–5, 2017.

[129] Y. Wang, J. Ramadani, and S. Wagner, "An exploratory study on applying a scrum development process for safety-critical systems," in *Product-Focused Software Process Improvement: 18th International Conference, PROFES 2017, Innsbruck, Austria, November 29–December 1, 2017, Proceedings 18*, pp. 324–340, Springer, 2017.

[130] T. Stralhane, T. Myklebust, and G. Hanssen, "The application of safe scrum to iec 61508 certifiable software," in *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference*, vol. 8, pp. 6052–6061, 2012.

[131] T. Stalhane and T. Myklebust, "The agile safety case," in *SAFECOMP Workshops*, vol. 9923 LNCS, pp. 5–16, 2016.

[132] T. Myklebust, T. Stalhane, and G. K. Hanssen, "Agile safety case and devops for the automotive industry," in *30th European Safety and Reliability Conference, ESREL and 15th Probabilistic Safety Assessment and Management Conference, PSAM*, 2020.

[133] T. Myklebust, T. Stalhane, and S. Wu, "Agile safety case for vehicle trial operations," in *16th International Conference on Probabilistic Safety Assessment and Management, PSAM*, 2022.

[134] M. Sivakumar, A. B. Belle, K. K. Shahandashthi, O. Odu, H. Hemmati, L. Briand, S. Kpodjedo, S. Wang, and O. Adesina, "I came, i saw, i certified: some perspectives on the safety assurance of cyber-physical systems," *submitted to IEEE Software*, 2023.

[135] H. Dezfuli, C. Everett, and A. Benjamin, "A "systems/case-based" approach to system safety," in *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference 2012, PSAM11 ESREL 2012*, vol. 7, pp. 5165–5174, 2012.

[136] V. Cassano, T. S. E. Maibaum, and S. Grigorova, "A (proto) logical basis for the notion of a structured argument in a safety case," in *Formal Methods and Software Engineering: 18th International Conference on Formal Engineering Methods, ICFEM 2016, Tokyo, Japan, November 14-18, 2016, Proceedings 18*, vol. 10009 LNCS, pp. 1–17, Springer Verlag, 2016.

[137] E. Denney and G. Pai, "A formal basis for safety case patterns," in *Computer Safety, Reliability, and Security: 32nd International Conference, SAFECOMP 2013, Toulouse, France, September 24-27, 2013. Proceedings 32*, vol. 8153 LNCS, pp. 21–32, 2013.

[138] R. Wang, J. Guiochet, and G. Motet, "A framework for assessing safety argumentation confidence," in *Software Engineering for Resilient Systems: 8th International Workshop, SERENE 2016, Gothenburg, Sweden, September 5-6, 2016, Proceedings 8*, vol. 9823 LNCS, pp. 3–12, Springer Verlag, 2016.

[139] R. D. Hawkins and T. P. Kelly, "A framework for determining the sufficiency of software safety assurance," in *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*, pp. 1–6, IET, 2012.

[140] G. Despotou, M. Ryan, T. N. Arvanitis, A. J. Rae, S. White, T. Kelly, and R. W. Jones, "A framework for synthesis of safety justification for digitally enabled healthcare services," *Digital health*, 2017.

[141] C. L. Lin, W. Shen, and S. Drager, "A framework to support generation and maintenance of an assurance case," in *Proceedings - 2016 IEEE 27th International Symposium on Software Reliability Engineering Workshops, ISSREW 2016*, pp. 21–24, Institute of Electrical and Electronics Engineers Inc., 2016.

[142] S. Yamamoto, "A knowledge integration approach of safety-critical software development and operation based on the method architecture," in *18th International Conference on Knowledge-Based and Intelligent Information Engineering Systems - KES2014*, vol. 35, pp. 1718–1727, Elsevier B.V., 2014.

[143] J. Birch, R. Rivett, I. Habli, B. Bradshaw, J. Botham, D. Higham, H. Monkhouse, and R. Palin, "A layered model for structuring automotive safety arguments," in *Proceedings - 2014 10th European Dependable Computing Conference, EDCC 2014*, pp. 178–181, IEEE Computer Society, 2014.

[144] T. Viger, L. Murphy, A. Di Sandro, R. Shahin, and M. Chechik, "A lean approach to building valid model-based safety arguments," in *Proceedings - 24th International Conference on Model-Driven Engineering Languages and Systems, MODELS 2021*, pp. 194–204, Institute of Electrical and Electronics Engineers Inc., 2021.

[145] E. Denney and G. Pai, "A lightweight methodology for safety case assembly," in *Computer Safety, Reliability, and Security: 31st International Conference, SAFECOMP 2012, Magdeburg, Germany, September 25-28, 2012. Proceedings 31*, pp. 1–12, Springer, 2012.

[146] S. Björnander, R. Land, P. Graydon, K. Lundqvist, and P. Conmy, "A method to formally evaluate safety case arguments against a system architecture model," in *Proceedings - 23rd IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2012*, pp. 337–342, 2012.

[147] I. vSljivo, B. Gallina, J. Carlson, H. Hansson, and S. Puri, "A method to generate reusable safety case argument-fragments from compositional safety analysis," *Journal of Systems and Software*, vol. 131, pp. 570–590, 2017.

[148] J. Guiochet, H. Quynh Anh, and M. Kaaniche, "A model for safety case confidence assessment," in *Computer Safety, Reliability and Security. 34th International Conference, SAFECOMP*, pp. 313–27, 2015.

[149] B. Gallina, "A model-driven safety certification method for process compliance," in *Proceedings - IEEE 25th International Symposium on Software Reliability Engineering Workshops, ISSREW 2014*, pp. 204–209, Institute of Electrical and Electronics Engineers Inc., 2014.

[150] A. Larrucea, J. Perez, and R. Obermaisser, "A modular safety case for an iec-61508 compliant generic cots multicore processor," in *Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015*, pp. 1788–1795, Institute of Electrical and Electronics Engineers Inc., 2015.

[151] A. Larrucea, J. Perez, I. Agirre, V. Brocal, and R. Obermaisser, "A modular safety case for an iec-61508 compliant generic hypervisor," in *Proceedings - 18th Euromicro Conference on Digital System Design, DSD 2015*, pp. 571–574, Institute of Electrical and Electronics Engineers Inc., 2015.

[152] M. Khalil, A. Prieto, and F. Hölzl, "A pattern-based approach towards the guided reuse of safety mechanisms in the automotive domain," vol. 8822, pp. 137–151, 2014.

[153] D. Nevsić, M. Nyberg, and B. Gallina, "A probabilistic model of belief in safety cases," *Safety science*, vol. 138, p. 105187, 2021.

[154] Y. Idmessaoud, D. Dubois, and J. Guiochet, "A qualitative counterpart of belief functions with application to uncertainty propagation in safety cases," in *International Conference on Belief Functions*, vol. 13506 LNAI, pp. 231–241, Springer Science and Business Media Deutschland GmbH, 2022.

[155] L. Feng, A. L. King, S. Chen, A. Ayoub, J. Park, N. Bezzo, O. Sokolsky, and I. Lee, "A safety argument strategy for pca closed-loop systems: A preliminary proposal," in *5th Workshop on Medical Cyber-Physical Systems*, vol. 36, pp. 94–99, 2014.

[156] T. Schmid, S. Schraufstetter, S. Wagner, and D. Hellhake, "A safety argumentation for fail-operational automotive systems in compliance with iso 26262," in *2019 4th International Conference on System Reliability and Safety (ICSRS)*, pp. 484–493, IEEE, 2019.

[157] A. Ayoub, B. Kim, I. Lee, and O. Sokolsky, "A safety case pattern for model-based development approach," in *NASA Formal Methods: 4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012. Proceedings 4*, pp. 141–146, Springer, 2012.

[158] Q. Liu, W. Zhang, X. Yue, and Q. Yang, "A safety-argument based method to predict system failure," in *Proceedings of IEEE 2012 Prognostics and System Health Management Conference, PHM-2012*, 2012.

[159] C. Menon and R. Alexander, "A safety-case approach to the ethics of autonomous vehicles," *Safety and reliability*, 2020.

[160] J. Birch, D. Blackburn, J. Botham, I. Habli, D. Higham, H. Monkhouse, G. Price, N. Ratiu, and R. Rivett in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*, vol. 12235 LNCS, pp. 408–414, Springer Science and Business Media Deutschland GmbH, 2020.

[161] Y. Luo, M. van den Brand, Z. Li, and A. K. Saberi, "A systematic approach and tool support for gsn-based safety case assessment," *Journal of Systems Architecture*, vol. 76, pp. 1–16, 2017.

[162] A. Ayoub, B. Kim, I. Lee, and O. Sokolsky, "A systematic approach to justifying sufficient confidence in software safety arguments," in *International Conference on Computer Safety, Reliability, and Security*, pp. 305–316, Springer, 2012.

[163] S. Vorapojpisut, "A v-model framework for the certification against the annex r of iec 60335-1: Class b appliances," in *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 30–35, Institute of Electrical and Electronics Engineers Inc., 2016.

[164] J. L. de la Vara, G. Génova, J. M. Álvarez Rodr´iguez, and J. Llorens, "An analysis of safety evidence management with the structured assurance case metamodel," *Computer Standards and Interfaces*, vol. 50, pp. 179–198, 2017.

[165] F. R. Ward and I. Habli, "An assurance case pattern for the interpretability of machine learning in safety-critical systems," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*, vol. 12235 LNCS, pp. 395–407, Springer Science and Business Media Deutschland GmbH, 2020.

[166] S. Yamamoto and Y. Matsuno, "An evaluation of argument patterns to reduce pitfalls of applying assurance case," in *ASSURE*, pp. 12–17, IEEE, 2013.

[167] S. Nair, N. Walkinshaw, T. Kelly, and J. L. de la Vara, "An evidential reasoning approach for assessing confidence in safety evidence," in *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 541–552, IEEE, 2015.

[168] X. Larrucea, P. González-Nalda, I. Etxeberria-Agiriano, M. C. Otero, and I. Calvo in *New Trends in Model and Data Engineering: MEDI 2018 International Workshops, DETECT, MEDI4SG, IWCFS, REMEDY, Marrakesh, Morocco, October 24–26, 2018, Proceedings 8*, vol. 929, pp. 167–180, Springer Verlag, 2018.

[169] C. L. Lin and W. Shen, "Applying safety case pattern to generate assurance cases for safety-critical systems," in *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, vol. 2015-January, pp. 255–262, IEEE Computer Society, 2015.

[170] M. Gleirscher and C. Carlan, "Arguing from hazard analysis in safety cases: A modular argument pattern," in *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, pp. 53–60, IEEE Computer Society, 2017.

[171] C. Cârlan, B. Gallina, S. Kacianka, and R. Breu, "Arguing on software-level verification techniques appropriateness," in *Computer Safety, Reliability, and Security: 36th International Conference, SAFECOMP 2017, Trento, Italy, September 13-15, 2017, Proceedings 36*, pp. 39–54, Springer, 2017.

[172] A. B. Hocking, J. Knight, M. A. Aiello, and S. Shiraishi, "Arguing software compliance with iso 26262," in *Proceedings - IEEE 25th International Symposium on Software Reliability Engineering Workshops, ISSREW 2014*, pp. 226–231, Institute of Electrical and Electronics Engineers Inc., 2014.

[173] S. Grigorova and T. S. E. Maibaum, "Argument evaluation in the context of assurance case confidence modeling," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 485–490, 2014.

[174] E. Denney and G. Pai, "Argument-based airworthiness assurance of small uas," in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2015.

[175] T. Yuan and T. Kelly, "Argument-based approach to computer system safety engineering," *International Journal of Critical Computer-Based Systems*, vol. 3, no. 3, pp. 151–167, 2012.

[176] J. Reich, J. Frey, E. Cioroaica, M. Zeller, and M. Rothfelder, "Argument-driven safety engineering of a generic infusion pump with digital dependability identities," in *Model-Based Safety and Assessment: 7th International Symposium, IMBSA 2020, Lisbon, Portugal, September 14–16, 2020, Proceedings 7*, vol. 12297 LNCS, pp. 19–33, 2020.

[177] J. L. de la Vara, G. Jiménez, R. Mendieta, and E. Parra, "Assessment of the quality of safety cases: A research preview," in *Requirements Engineering: Foundation for Software Quality: 25th International Working Conference, REFSQ 2019, Essen, Germany, March 18–21, 2019, Proceedings 25*, vol. 11412 LNCS, pp. 124–131, Springer Verlag, 2019.

[178] C. Picardi, C. Paterson, R. Hawkins, R. Calinescu, and I. Habli, "Assurance argument patterns and processes for machine learning in safety-related systems," in *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI 2020)*, vol. 2560, pp. 23–30, CEUR-WS, 2020.

[179] Y. Zhang, B. Larson, and J. Hatcliff, "Assurance case considerations for interoperable medical systems," in *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerras, Sweden, September 18, 2018, Proceedings 37*, vol. 11094 LNCS, pp. 42–48, Springer Verlag, 2018.

[180] A. Schwierz and H. Forsberg, "Assurance case to structure cots hardware component assurance for safety-critical avionics," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pp. 1–10, 2018.

[181] E. Asaadi, S. Beland, A. Chen, E. Denney, D. Margineantu, M. Moser, G. Pai, J. Paunicka, D. Stuart, and H. Yu, "Assured integration of machine learning-based autonomy on aviation platforms," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pp. 1–10, 2020.

[182] E. Denney, G. Pai, R. Berthold, M. Fladeland, B. Storms, and M. Sumich, "Assuring ground-based detect and avoid for uas operations," in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pp. 6A11–6A116, Institute of Electrical and Electronics Engineers Inc., 2014.

[183] P. Conmy and I. Bate, "Assuring safety for component based software engineering," in *Proceedings - 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering, HASE 2014*, pp. 121–128, IEEE Computer Society, 2014.

[184] J. Reich, M. Zeller, and D. Schneider, "Automated evidence analysis of safety arguments using digital dependability identities," in *Computer Safety, Reliability, and Security: 38th International Conference, SAFECOMP 2019, Turku, Finland, September 11–13, 2019, Proceedings 38*, vol. 11698 LNCS, pp. 254–268, Springer Verlag, 2019.

[185] C. Hartsell, N. Mahadevan, A. Dubey, and G. Karsai, "Automated method for assurance case construction from system design models," in *2021 5th International Conference on System Reliability and Safety (ICSRS)*, pp. 230–239, 2021.

[186] E. Armengaud, "Automated safety case compilation for product-based argumentation," *Embedded Real Time Software and Systems*, 2014.

[187] C. Cârlan, L. Gauerhof, B. Gallina, and S. Burton, "Automating safety argument change impact analysis for machine learning components," in *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC*, pp. 43–53, IEEE Computer Society, 2022.

[188] E. Denney and G. Pai, "Automating the assembly of aviation safety cases," *IEEE Transactions on Reliability*, vol. 63, no. 4, pp. 830–849, 2014.

[189] G. Macher, H. Sporer, and C. Kreiner, "Automotive safety case pattern," in *Proceedings of the 19th European Conference on Pattern Languages of Programs*, 2014.

[190] Y. Idmessaoud, D. Dubois, and J. Guiochet, "Belief functions for safety arguments confidence estimation: A comparative study," in *Scalable Uncertainty Management: 14th International Conference, SUM 2020, Bozen-Bolzano, Italy, September 23–25, 2020, Proceedings 14*, vol. 12322 LNAI, pp. 141–155, Springer Science and Business Media Deutschland GmbH, 2020.

[191] B. P. Williams, R. Clothier, N. Fulton, X. Lin, S. Johnson, and K. Cox, "Building the safety case for uas operations in support of natural disaster response," in *AIAA AVIATION 2014 -14th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics Inc., 2014.

[192] A. Wassyng, N. K. Singh, M. Geven, N. Proscia, H. Wang, M. Lawford, and T. Maibaum, "Can product-specific assurance case templates be used as medical device standards?," *IEEE Design Test*, vol. 32, no. 5, pp. 45–55, 2015.

[193] C. Carlan, D. Petrisor, B. Gallina, and H. Schoenhaar, "Checkable safety cases: Enabling automated consistency checks between safety work products," in *Proceedings - 2020 IEEE 31st International Symposium on Software Reliability Engineering Workshops, ISSREW 2020*, pp. 295–302, Institute of Electrical and Electronics Engineers Inc., 2020.

[194] C. Hirata and S. Nadjm-Tehrani, "Combining gsn and stpa for safety arguments," in *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, Turku, Finland, September 10, 2019, Proceedings 38*, vol. 11699 LNCS, pp. 5–15, Springer Verlag.

[195] E. Denney and G. Pai, "Composition of safety argument patterns," in *Computer Safety, Reliability, and Security: 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016, Proceedings 35*, pp. 51–63, Springer, 2016.

[196] T. Yuan, T. Kelly, and T. Xu, "Computer-assisted safety argument review–a dialectics approach," *Argument & Computation*, vol. 6, no. 2, pp. 130–148, 2015.

[197] S. Burton, L. Gauerhof, B. B. Sethy, I. Habli, and R. Hawkins, "Confidence arguments for evidence of performance in machine learning for highly automated driving functions," in *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, pp. 365–377, Springer, 2019.

[198] R. Wang, J. Guiochet, and G. Motet, "Confidence assessment framework for safety arguments," in *Computer Safety, Reliability, and Security: 36th International Conference, SAFECOMP 2017, Trento, Italy, September 13-15, 2017, Proceedings 36*, pp. 55–68, Springer, 2017.

[199] A. Groza and N. Marc, "Consistency checking of safety arguments in the goal structuring notation standard," in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 59–66, IEEE, 2014.

[200] D. Nevsić, M. Nyberg, and B. Gallina, "Constructing product-line safety cases from contract-based specifications," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 2022–2031, 2019.

[201] A. Ray and R. Cleaveland, "Constructing safety assurance cases for medical devices," in *2013 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, pp. 40–45, IEEE, 2013.

[202] F. Warg, H. Blom, J. Borg, and R. Johansson, "Continuous deployment for dependable systems with continuous assurance cases," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 318–325, IEEE, 2019.

[203] A. L. J. Dominguez, B. G. Partridge, and J. J. Joyce, "Creating safety assurance cases for rebreather systems," in *2013 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, pp. 34–39, IEEE, 2013.

[204] T. Chowdhury, A. Wassyng, R. F. Paige, and M. Lawford, "Criteria to systematically evaluate (safety) assurance cases," in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 380–390, IEEE, 2019.

[205] T. A. Beyene and C. Carlan, "Cybergsn: a semi-formal language for specifying safety cases," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 63–66, IEEE, 2021.

[206] O. Jaradat and I. Bate, "Deriving hierarchical safety contracts," in *2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 119–128, IEEE, 2015.

[207] B. Gallina and L. Provenzano, "Deriving reusable process-based arguments from process models in the context of railway safety standards," in *Ada User Journal*, vol. 36, pp. 237–241, Ada-Europe, 2015.

[208] B. Gallina, E. Gómez-Mart´inez, and C. B. Earle, "Deriving safety case fragments for assessing mbasafe's compliance with en 50128," in *Software Process Improvement and Capability Determination: 16th International Conference, SPICE 2016, Dublin, Ireland, June 9-10, 2016, Proceedings 16*, pp. 3–16, Springer, 2016.

[209] I. Sljivo, O. Jaradat, I. Bate, and P. Graydon, "Deriving safety contracts to support architecture design of safety critical systems," in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, pp. 126–133, IEEE, 2015.

[210] B. Gallina and A. Andrews, "Deriving verification-related means of compliance for a model-based testing process," in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2016.

[211] Y. Jia, T. Lawton, S. White, and I. Habli, "Developing a safety case for electronic prescribing," in *Studies in Health Technology and Informatics*, vol. 264, pp. 629–633, IOS Press, 2019.

[212] A. Carr, D. Murphy, I. Dugdale, A. Dyson, I. Habli, and R. Robinson, "Developing the safety case for medipi: An open-source platform for self management," *Studies in Health Technology and Informatics*, vol. 235, pp. 78–82, 2017.

[213] Y. Luo, Z. Li, and M. Van Den Brand, "Development of a safety case editor with assessment features," in *Proceedings - 2016 Workshop on Automotive Systems/Software Architectures, WASA 2016*, pp. 10–13, Institute of Electrical and Electronics Engineers Inc., 2016.

[214] R. Wang, J. Guiochet, G. Motet, and W. Schön, "D-s theory for argument confidence assessment," in *Belief Functions: Theory and Applications: 4th International Conference, BELIEF 2016, Prague, Czech Republic, September 21-23, 2016, Proceedings 4*, vol. 9861 LNAI, pp. 190–200, Springer Verlag.

[215] S. Diemert and J. Joyce, "Eliminative argumentation for arguing system safety - a practitioner's experience," in *SYSCON 2020 - 14th Annual IEEE International Systems Conference, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2020.

[216] B. Gallina, S. Kashiyarandi, K. Zugsbratl, and A. Geven, "Enabling cross-domain reuse of tool qualification certification artefacts," in *Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops: DEVVARTS. Florence, Italy, September 8-9, 2014. Proceedings 33*, vol. 8696 LNCS, pp. 255–266, Springer Verlag, 2014.

[217] J. Reich, D. Schneider, I. Sorokos, Y. Papadopoulos, T. Kelly, R. Wei, E. Armengaud, and C. Kaypmaz, "Engineering of runtime safety monitors for cyber-physical systems with digital dependability identities," in *Computer Safety, Reliability, and Security: 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, September 16–18, 2020, Proceedings 39*, vol. 12234 LNCS, pp. 3–17, Springer.

[218] C. Cârlan and B. Gallina, "Enhancing state-of-the-art safety case patterns to support change impact analysis," in *30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM 2020*, pp. 4658–4665, Research Publishing Services, 2020.

[219] E. Denney and G. Pai, "Evidence arguments for using formal methods in software certification," in *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 375–380, IEEE, 2013.

[220] C. Cârlan, S. Barner, A. Diewald, A. Tsalidis, and S. Voss, "Explicitcase: integrated model-based development of system and safety cases," in *Computer Safety, Reliability, and Security: SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS, Trento, Italy, September 12, 2017, Proceedings 36*, pp. 52–63, Springer, 2017.

[221] C. Cârlan, V. Nigam, S. Voss, and A. Tsalidis, "Explicitcase: tool-support for creating and maintaining assurance arguments integrated with system models," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 330–337, IEEE, 2019.

[222] Y. Prokhorova, L. Laibinis, and E. Troubitsyna, "Facilitating construction of safety cases from formal models in event-b," *Information and Software Technology*, vol. 60, pp. 51–76, 2015.

[223] O. Jaradat, I. Bate, and S. Punnekkat, "Facilitating the maintenance of safety cases," in *Current Trends in Reliability, Availability, Maintainability and Safety: An Industry Perspective*, pp. 349–371, Springer, 2016.

[224] C. Cârlan and D. Ratiu, "Fasten. safe: A model-driven engineering tool to experiment with checkable assurance cases," in *Computer Safety, Reliability, and Security: 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, September 16–18, 2020, Proceedings 39*, pp. 298–306, Springer, 2020.

[225] E. Denney, G. Pai, and I. Whiteside, "Formal foundations for hierarchical safety cases," in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, pp. 52–59, IEEE, 2015.

[226] A. Iliasov, D. Taylor, L. Laibinis, and A. Romanovsky, "Formal verification of railway interlocking and its safety case," in *Safety-critical Systems Symposium 2022*, Newcastle University, 2022.

[227] L. Laibinis, E. Troubitsyna, Y. Prokhorova, A. Iliasov, and A. Romanovsky, "From requirements engineering to safety assurance: refinement approach," in *Dependable Software Engineering: Theories,*

*Tools, and Applications: First International Symposium, SETTA 2015, Nanjing, China, November 4-6, 2015, Proceedings 1*, pp. 201–216, Springer, 2015.

[228] K. P. Woodham, P. Graydon, N. K. Borer, K. V. Papathakis, T. Stoia, and C. Balan, "Fueleap model-based system safety analysis," in *2018 Aviation Technology, Integration, and Operations Conference*, p. 3362, 2018.

[229] F. Zeng, M. Lu, and D. Zhong, "General development framework and its application method for software safety case.," *J. Softw.*, vol. 8, no. 12, pp. 3262–3268, 2013.

[230] N. Annable, T. Chiang, M. Lawford, R. F. Paige, and A. Wassyng, "Generating assurance cases using workflow+ models," in *International Conference on Computer Safety, Reliability, and Security*, pp. 97–110, Springer, 2022.

[231] I. Sljivo, B. Gallina, J. Carlson, and H. Hansson, "Generation of safety case argument-fragments from safety contracts," in *Computer Safety, Reliability, and Security: 33rd International Conference, SAFECOMP 2014, Florence, Italy, September 10-12, 2014. Proceedings 33*, pp. 170–185, Springer, 2014.

[232] D. Zapata, I. Pederson, and S. Keane, "Geohazard management approach within safety case," in *International Pipeline Conference*, vol. 51876, p. V002T07A025, American Society of Mechanical Engineers, 2018.

[233] M. Chelouati, A. Boussif, J. Beugin, and E.-M. El Koursi, "Graphical safety assurance case using goal structuring notation (gsn)—challenges, opportunities and a framework for autonomous trains," *Reliability Engineering & System Safety*, vol. 230, p. 108933, 2023.

[234] C.-F. Nicolas, F. Eizaguirre, A. Larrucea, S. Barner, F. Chauvel, G. Sagardui, and J. Perez, "Gsn support of mixed-criticality systems certification," in *Computer Safety, Reliability, and Security: SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS, Trento, Italy, September 12, 2017, Proceedings 36*, pp. 157–172, Springer, 2017.

[235] E. Denney, G. Pai, and J. Pohl, "Heterogeneous aviation safety cases: Integrating the formal and the non-formal," in *2012 IEEE 17th International Conference on Engineering of Complex Computer Systems*, pp. 199–208, IEEE, 2012.

[236] E. Denney, G. Pai, and I. Whiteside, "Hierarchical safety cases," in *NASA Formal Methods: 5th International Symposium, NFM 2013, Moffett Field, CA, USA, May 14-16, 2013. Proceedings 5*, pp. 478–483, Springer, 2013.

[237] Q. A. Do Hoang, J. Guiochet, D. Powell, and M. Kaâniche, "Human-robot interactions: Model-based risk analysis and safety case construction," in *Embedded Real Time Software and Systems (ERTS2 2012)*, 2012.

[238] R. Dardar, B. Gallina, A. Johnsen, K. Lundqvist, and M. Nyberg, "Industrial experiences of building a safety case in compliance with iso 26262," in *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, pp. 349–354, IEEE, 2012.

[239] C. Cârlan, T. A. Beyene, and H. Ruess, "Integrated formal methods for constructing assurance cases," in *2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 221–228, 2016.

[240] K. Lukasiewicz and J. Górski, "Introducing agile practices into development processes of safety critical software," in *Proceedings of the 19th International Conference on Agile Software Development: Companion*, vol. Part F147763, Association for Computing Machinery, 2018.

[241] G. Despotou, S. White, T. Kelly, and M. Ryan, "Introducing safety cases for health it," in *2012 4th International Workshop on Software Engineering in Health Care, SEHC 2012 - Proceedings*, pp. 44–50, 2012.

[242] I. Ibarra, S. Hartley, S. Crozier, and D. Ward, "Iso 26262 concept phase safety argument for a complex item," in *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*, 2012.

[243] K. Kakimoto, K. Sasaki, H. Umeda, and Y. Ueda, "Ivv case: Empirical study of software independent verification and validation based on safety case," in *Proceedings - 2017 IEEE 28th International Symposium on Software Reliability Engineering Workshops, ISSREW 2017*, pp. 32–35, Institute of Electrical and Electronics Engineers Inc., 2017.

[244] J. M. Brain, "Learning from experience – how can we produce a nuclear safety case to outlast the station?," in *9th IET International Conference on System Safety and Cyber Security (2014)*, pp. 1–6, 2014.

[245] Y. Prokhorova and E. Troubitsyna, "Linking modelling in event-b with safety cases," in *4th International Workshop on Software Engineering for Resilient Systems, SERENE 2012*, vol. 7527 LNCS, pp. 47–62, 2012.

[246] K. Taguchi, S. Daisuke, H. Nishihara, and T. Takai, "Linking traceability with gsn," in *Proceedings - IEEE 25th International Symposium on Software Reliability Engineering Workshops, ISSREW 2014*, pp. 192–197, Institute of Electrical and Electronics Engineers Inc., 2014.

[247] C. Cârlan, "Living safety arguments for open systems," in *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 120–123, IEEE, 2017.

[248] I. Sorokos, Y. Papadopoulos, and L. Bottaci, "Maintaining safety arguments via automatic allocation of safety requirements," *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 25–30, 2016.

[249] R. Nevalainen, A. Ruiz, and T. Varkoi, "Making software safety assessable and transparent," in *20th European Conference on System and Software Process Improvement, EuroSPI 2013*, vol. 364 CCIS, pp. 1–12, Springer Verlag, 2013.

[250] C. L. Lin, W. Shen, S. Drager, and B. Cheng, "Measure confidence of assurance cases in safety-critical domains," in *2018 IEEE/ACM 40th International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE-NIER)*, pp. 13–16, 2018.

[251] R. Boring and N. Lau, "Measurement sufficiency versus completeness: Integrating safety cases into verification and validation in nuclear control room modernization," in *Advances in Intelligent Systems and Computing*, vol. 495, pp. 79–90, Springer Verlag, 2017.

[252] A. Di Sandro, G. Selim, R. Salay, T. Viger, M. Chechik, and S. Kokaly, "Mmint-a 2.0: Tool support for the lifecycle of model-based safety artifacts," in *Proceedings - 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings*, pp. 71–75, Association for Computing Machinery, Inc, 2020.

[253] C. Hartsell, N. Mahadevan, S. Ramakrishna, A. Dubey, T. Bapty, T. Johnson, X. Koutsoukos, J. Sztipanovits, and G. Karsai, "Model-based design for cps with learning-enabled components," in *DESTION 2019 - Proceedings of the Workshop on Design Automation for CPS and IoT*, pp. 1–9, Association for Computing Machinery, Inc, 2019.

[254] E. Denney, G. Pai, and I. Whiteside, "Model-driven development of safety architectures," in *Proceedings - ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems, MODELS 2017*, pp. 156–166, Institute of Electrical and Electronics Engineers Inc., 2017.

[255] R. Wang, J. Guiochet, G. Motet, and W. Schön, "Modelling confidence in railway safety case," *Safety Science*, vol. 110, pp. 286–299, 2018.

[256] A. Larrucea, I. Martinez, C. F. Nicolas, J. Perez, and R. Obermaisser, "Modular development and certification of dependable mixed-criticality systems," in *Proceedings - 20th Euromicro Conference on Digital System Design, DSD 2017*, pp. 419–426, Institute of Electrical and Electronics Engineers Inc., 2017.

[257] A. Larrucea, I. Martinez, C. F. Nicolas, J. Perez, and R. Obermaisser, "Modular development and certification of dependable mixed-criticality systems," in *2017 Euromicro Conference on Digital System Design (DSD)*, pp. 419–426, IEEE, 2017.

[258] H. Agarwal, R. Dorociak, and A. Rettberg, "On safety assurance case for deep learning based image classification in highly automated driving," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1472–1477, IEEE, 2021.

[259] L. Gauerhof, R. Gansch, C. Heinzemann, M. Woehrle, and A. Heyl, "On the necessity of explicit artifact links in safety assurance cases for machine learning," in *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 23–28, IEEE, 2021.

[260] E. Denney, G. Pai, and I. Habli, "Perspectives on software safety case development for unmanned aircraft," in *Proceedings of the International Conference on Dependable Systems and Networks*, 2012.

[261] B. Gallina and M. Nyberg, "Pioneering the creation of iso 26262-compliant oslc-based safety cases," in *Proceedings - 2017 IEEE 28th International Symposium on Software Reliability Engineering Workshops, ISSREW 2017*, pp. 325–330, Institute of Electrical and Electronics Engineers Inc., 2017.

[262] V. Katta, T. Stalhane, and C. Raspotnig, "Presenting a traceability based approach for safety argumentation," in *Safety, Reliability and Risk Analysis: Beyond the Horizon - Proceedings of the European Safety and Reliability Conference, ESREL 2013*, pp. 2037–2046, shers, 2013.

[263] M. Napolano, F. Machida, R. Pietrantuono, and D. Cotroneo, "Preventing recurrence of industrial control system accident using assurance case," in *2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 182–189, IEEE, 2015.

[264] T. Chowdhury, C.-W. Lin, B. Kim, M. Lawford, S. Shiraishi, and A. Wassyng, "Principles for systematic development of an assurance case template from iso 26262," in *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 69–72, IEEE, 2017.

[265] B. Gallina, E. Gómez-Mart´inez, and C. Benac-Earle, "Promoting mba in the rail sector by deriving process-related evidence via mdsafecer," *Computer Standards & Interfaces*, vol. 54, pp. 119–128, 2017.

[266] Y. Idmessaoud, D. Dubois, and J. Guiochet, "Quantifying confidence of safety cases with belief functions," in *Belief Functions: Theory and Applications: 6th International Conference, BELIEF 2021, Shanghai, China, October 15–19, 2021, Proceedings 6*, pp. 269–278, Springer, 2021.

[267] S. Nair, N. Walkinshaw, and T. Kelly, "Quantifying uncertainty in safety cases using evidential reasoning," in *Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops: ASCoMS, DECSoS, DEVVARTS, ISSE, ReSA4CI, SASSUR. Florence, Italy, September 8-9, 2014. Proceedings 33*, pp. 413–418, Springer, 2014.

[268] A. Di Sandro, S. Kokaly, R. Salay, and M. Chechik, "Querying automotive system models and safety artifacts with mmint and viatra," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 2–11, IEEE, 2019.

[269] E. Denney, D. Naylor, and G. Pai, "Querying safety cases," in *Computer Safety, Reliability, and Security: 33rd International Conference, SAFECOMP 2014, Florence, Italy, September 10-12, 2014. Proceedings 33*, pp. 294–309, Springer, 2014.

[270] P. Graydon and I. Bate, "Realistic safety cases for the timing of systems," *The Computer Journal*, vol. 57, no. 5, pp. 759–774, 2014.

[271] L. Duan, S. Rayadurgam, M. Heimdahl, O. Sokolsky, and I. Lee, "Representation of confidence in assurance cases using the beta distribution," in *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 86–93, IEEE, 2016.

[272] R. R. Lutz, "Requirements engineering for safety-critical molecular programs," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pp. 302–308, IEEE, 2022.

[273] L. Sun, N. Silva, and T. Kelly, "Rethinking of strategy for safety argument development," in *Computer Safety, Reliability, and Security: SAFECOMP 2014 Workshops: SASSUR. Florence, Italy, September 8-9, 2014. Proceedings 33*, pp. 384–395, Springer, 2014.

[274] S. Guarro, M. K. Yau, U. Ozguner, T. Aldemir, A. Kurt, M. Hejase, and M. Knudson, "Risk informed safety case framework for unmanned aircraft system flight software certification," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, p. 0910, 2017.

[275] D. Feng and C. Eyster, "Risk-based requirements management framework with applications to assurance cases," in *2013 IEEE Aerospace Conference*, pp. 1–11, IEEE, 2013.

[276] P. Koopman and B. Osyk, "Safety argument considerations for public road testing of autonomous vehicles," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, no. 2019-01-0123, pp. 512–523, 2019.

[277] S. Wang, Z. Li, X. He, K. Sheng, J. Zhou, and W. Wu, "Safety argument pattern language of safety-critical software," in *2020 7th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 459–467, IEEE, 2020.

[278] H. Fujino, N. Kobayashi, and S. Shirasaka, "Safety assurance case description method for systems incorporating off-operational machine learning and safety device," in *INCOSE International Symposium*, vol. 29, pp. 152–164, Wiley Online Library, 2019.

[279] S. Burton, I. Kurzidem, A. Schwaiger, P. Schleiss, M. Unterreiner, T. Graeber, and P. Becker, "Safety assurance of machine learning for chassis control functions," in *Computer Safety, Reliability, and Security: 40th International Conference, SAFECOMP 2021, York, UK, September 8–10, 2021, Proceedings 40*, pp. 149–162, Springer, 2021.

[280] R. Wang, J. Guiochet, G. Motet, and W. Schön, "Safety case confidence propagation based on dempster–shafer theory," *International Journal of Approximate Reasoning*, vol. 107, pp. 46–64, 2019.

[281] M. F. Inzaghi, D. M. Rosyid, W. L. Dhanistha, *et al.*, "Safety case development for risk management of offshore pipeline," in *IOP Conference Series: Earth and Environmental Science*, vol. 698, p. 012043, IOP Publishing, 2021.

[282] M. Standish, H. Auld, P. Caseley, and M. Hadley, "Safety case development: a process to implement the safety three-layered framework," 2014.

[283] A. Ruiz, P. Barbosa, Y. Medeiros, and H. Espinoza, "Safety case driven development for medical devices," in *Computer Safety, Reliability, and Security: 34th International Conference, SAFECOMP 2015, Delft, The Netherlands, September 23-25, 2015, Proceedings 34*, pp. 183–196, Springer, 2015.

[284] J.-E. Holmberg, P. Bishop, S. Guerra, and N. Thuy, "Safety case framework to provide justifiable reliability numbers for software systems," in *Proceedings of 11th International Probabilistic Safety Assessment and Management Conference and The Annual European Safety and Reliability Conference*, 2012.

[285] S. Kokaly, R. Salay, M. Chechik, M. Lawford, and T. Maibaum, "Safety case impact assessment in automotive software systems: An improved model-based approach," in *Computer Safety, Reliability, and Security: 36th International Conference, SAFECOMP 2017, Trento, Italy, September 13-15, 2017, Proceedings 36*, vol. 10488 LNCS, pp. 69–85, Springer Verlag, 2017.

[286] F. I. Pérez, "Safety case process in cuba: Transition from theory to practice," *Process Safety and Environmental Protection*, vol. 137, pp. 169–176, 2020.

[287] E. Mirzaei, C. Thomas, and M. Conrad, "Safety cases for adaptive systems of systems: state of the art and current challenges," in *European Dependable Computing Conference*, pp. 127–138, Springer, 2020.

[288] E. Mirzaei, C. Thomas, and M. Conrad, "Safety cases for adaptive systems of systems: State of the art and current challenges," in *16th European Dependable Computing Conference, EDCC 2020*, vol. 1279 CCIS, pp. 127–138, Springer Science and Business Media Deutschland GmbH, 2020.

[289] E. Denney and G. Pai, "Safety considerations for uas ground-based detect and avoid," in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2016.

[290] S. Nair, J. L. de la Vara, A. Melzi, G. Tagliaferri, L. De-La-Beaujardiere, and F. Belmonte, "Safety evidence traceability: Problem analysis and model," in *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings 20*, pp. 309–324, Springer, 2014.

[291] E. Heikkilä, R. Tuominen, R. Tiusanen, J. Montewka, and P. Kujala, "Safety qualification process for an autonomous ship prototype - a goal-based safety case approach," in *Marine Navigation - Proceedings of the International Conference on Marine Navigation and Safety of Sea Transportation, TRANSNAV 2017*, pp. 365–370, CRC Press/Balkema, 2017.

[292] D. Ratiu, M. Zeller, and L. Killian, "Safety.lab: Model-based domain specific tooling for safety argumentation," in *3rd International Workshop on Assurance Cases for Software-Intensive Systems ASSURE 2015*, vol. 9338, pp. 72–82, Springer Verlag, 2015.

[293] M. A. Meyer, S. Silberg, C. Granrath, C. Kugler, L. Wachtmeister, B. Rumpe, S. Christiaens, and J. Andert, "Scenario- and model-based systems engineering procedure for the sotif-compliant design of automated driving functions," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1599–1604, 2022.

[294] M. A. Aiello, A. B. Hocking, J. Knight, and J. Rowanhill, "Sct: a safety case toolkit," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 216–219, IEEE, 2014.

[295] F. Zeng, M. Lu, and D. Zhong, "Software safety certification framework based on safety case," in *Proceedings - 2012 International Conference on Computer Science and Service System, CSSS 2012*, pp. 566–569, 2012.

[296] G. Schwalbe, B. Knie, T. Sämann, T. Dobberphul, L. Gauerhof, S. Raafatnia, and V. Rocco, "Structuring the safety argumentation for deep neural network based perception in automotive applications," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*, pp. 383–394, Springer, 2020.

[297] C.-L. Lin, W. Shen, and R. Hawkins, "Support for safety case generation via model transformation," *SIGBED Rev.*, vol. 14, no. 2, p. 44–52, 2017.

[298] J. Górski, A. Jarzkebowicz, J. Miler, M. Witkowicz, J. Czyżnikiewicz, and P. Jar, "Supporting assurance by evidence-based argument services," in *Computer Safety, Reliability, and Security: SAFECOMP 2012 Workshops: Sassur, ASCoMS, DESEC4LCCI, ERCIM/EWICS, IWDE, Magdeburg, Germany, September 25-28, 2012. Proceedings 31*, pp. 417–426, Springer, 2012.

[299] A. L. De Oliveira, R. T. Braga, P. C. Masiero, Y. Papadopoulos, I. Habli, and T. Kelly, "Supporting the automated generation of modular product line safety cases," in *Theory and Engineering of Complex Systems and Dependability: Proceedings of the Tenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, June 29–July 3 2015, Brunów, Poland*, pp. 319–330, Springer, 2015.

[300] H. Bagheri, E. Kang, and N. Mansoor, "Synthesis of assurance cases for software certification," in *2020 IEEE/ACM 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 61–64, 2020.

[301] T. Chowdhury, A. Wassyng, R. F. Paige, and M. Lawford, "Systematic evaluation of (safety) assurance cases," in *Computer Safety, Reliability, and Security: 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, September 16–18, 2020, Proceedings 39*, pp. 18–33, Springer, 2020.

[302] O. Jaradat and I. Bate, "Systematic maintenance of safety cases to reduce risk," in *Computer Safety, Reliability, and Security: SAFECOMP 2016 Workshops, ASSURE, DECSoS, SASSUR, and TIPS, Trondheim, Norway, September 20, 2016, Proceedings 35*, pp. 17–29, Springer, 2016.

[303] Y. Matsuno, F. Ishikawa, and S. Tokumoto, "Tackling uncertainty in safety assurance for machine learning: continuous argument engineering with attributed tests," in *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, pp. 398–404, Springer, 2019.

[304] T. Stralhane and T. Myklebust, "The agile safety case," p. pp 5–16, 2016.

[305] V. Cassano and T. S. Maibaum, "The definition and assessment of a safety argument," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 180–185, IEEE, 2014.

[306] J. Yang, M. Ward, and J. Akhtar, "The development of safety cases for an autonomous vehicle: A comparative study on different methods," in *SAE 2017 Intelligent and Connected Vehicles Symposium, ICVS 2017*, vol. Part F129886, SAE International, 2017.

[307] M. Sujan, P. Spurgeon, M. Cooke, A. Weale, P. Debenham, and S. Cross, "The development of safety cases for healthcare services: practical experiences, opportunities and challenges," *Reliability Engineering & System Safety*, vol. 140, pp. 200–207, 2015.

[308] G. Ellor, "The e-safetycase - electronic or effortless?," in *36th Center for Chemical Process Safety International Conference, CCPS 2021 - Topical Conference at the 2021 AIChE Spring Meeting and 17th Global Congress on Process Safety*, pp. 85–103, AIChE, 2021.

[309] T. Viger, L. Murphy, A. Di Sandro, C. Menghi, R. Shahin, and M. Chechik, "The foremost approach to building valid model-based safety arguments," *Software and Systems Modeling*, 2022.

[310] R. Salay, K. Czarnecki, H. Kuwajima, H. Yasuoka, T. Nakae, V. Abdelzad, C. Huang, M. Kahn, and V. D. Nguyen, "The missing link: Developing a safety case for perception components in automated driving," in *SAE 2022 Annual World Congress Experience, WCX 2022*, 2021.

[311] E. Denney, G. Pai, and I. Whiteside, "The role of safety architectures in aviation safety cases," *Reliability Engineering and System Safety*, vol. 191, 2019.

[312] M. Standish, H. J. Auld, P. R. Caseley, and M. J. Hadley, "The safety three-layer framework: a case study," in *9th IET International Conference on System Safety and Cyber Security (2014)*, pp. 1–8, 2014.

[313] P. J. Graydon, "Towards a clearer understanding of context and its role in assurance argument confidence," in *Computer Safety, Reliability, and Security: 33rd International Conference, SAFECOMP 2014, Florence, Italy, September 10-12, 2014. Proceedings 33*, pp. 139–154, Springer, 2014.

[314] E. Denney and G. Pai, "Towards a formal basis for modular safety cases," in *Computer Safety, Reliability, and Security: 34th International Conference, SAFECOMP 2015, Delft, The Netherlands, September 23-25, 2015, Proceedings 34*, pp. 328–343, Springer, 2015.

[315] J. A. McDermid, Y. Jia, and I. Habli, "Towards a framework for safety assurance of autonomous systems," in *Artificial Intelligence Safety 2019*, pp. 1–7, CEUR Workshop Proceedings, 2019.

[316] F. Geissler, S. Qutub, S. Roychowdhury, A. Asgari, Y. Peng, A. Dhamasia, R. Graefe, K. Pattabiraman, and M. Paulitsch, "Towards a safety case for hardware fault tolerance in convolutional neural networks using activation range supervision," in *CEUR Workshop Proceedings*, vol. 2916, CEUR-WS, 2021.

[317] R. Eastwood, T. P. Kelly, R. D. Alexander, and E. Landre, "Towards a safety case for runtime risk and uncertainty management in safety-critical systems," in *IET Conference Publications*, 2013.

[318] B. Gallina, K. Padira, and M. Nyberg, "Towards an iso 26262-compliant oslc-based tool chain enabling continuous self-assessment," in *Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016*, pp. 199–204, Institute of Electrical and Electronics Engineers Inc.

[319] M. Brito, "Towards building a safety case for marine unmanned surface vehicles: a bayesian perspective," 2017.

[320] R. Shahin, S. Kokaly, and M. Chechik, "Towards certified analysis of software product line safety cases," in *Computer Safety, Reliability, and Security: 40th International Conference, SAFECOMP 2021, York, UK, September 8–10, 2021, Proceedings 40*, vol. 12852 LNCS, pp. 130–145, Springer Science and Business Media Deutschland GmbH, 2021.

[321] S. Alajrami, B. Gallina, I. Sljivo, A. Romanovsky, and P. Isberg, "Towards cloud-based enactment of safety-related processes," in *Computer Safety, Reliability, and Security: 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016, Proceedings 35*, pp. 309–321, Springer, 2016.

[322] B. Gallina, "Towards enabling reuse in the context of safety-critical product lines," in *Proceedings - 5th International Workshop on Product Line Approaches in Software Engineering, PLEASE 2015*, pp. 15–18, Institute of Electrical and Electronics Engineers Inc., 2015.

[323] A. Wardziński and A. Jarzkebowicz, "Towards safety case integration with hazard analysis for medical devices," in *Computer Safety, Reliability, and Security: SAFECOMP 2016 Workshops, ASSURE, DECSoS, SASSUR, and TIPS, Trondheim, Norway, September 20, 2016, Proceedings 35*, pp. 87–98, Springer, 2016.

[324] Y. Idmessaoud, D. Dubois, and J. Guiochet, "Uncertainty elicitation and propagation in gsn models of assurance cases," in *International Conference on Computer Safety, Reliability, and Security*, pp. 111–125, Springer, 2022.

[325] A. Mjeda and G. Botterweck, "Uncertainty entangled; modelling safety assurance cases for autonomous systems," *Electronic Communications of the EASST*, vol. 79, pp. 1–10, 2020.

[326] A. Wardziński and P. Jones, "Uniform model interface for assurance case integration with system models," in *International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2017 and 5th International Workshop on Assurance Cases for Software-Intensive Systems, ASSURE 2017*, vol. 10489 LNCS, pp. 39–51, Springer Verlag.

[327] D. Pissoort, T. Bultinck, J. Boydens, and J. Catrysse, "Use of the goal structuring notation (gsn) as generic notation for an 'emc assurance case'," in *EMC Europe 2019 - 2019 International Symposium on Electromagnetic Compatibility*, Institute of Electrical and Electronics Engineers Inc., 2019.

[328] M. Kläs, R. Adler, L. Jöckel, J. Groß, and J. Reich, "Using complementary risk acceptance criteria to structure assurance cases for safety-critical ai components," in *2021 Workshop on Artificial Intelligence Safety, AISafety 2021*, vol. 2916, CEUR-WS, 2020.

[329] F. Zeng, M. Lu, and D. Zhong, "Using d-s evidence theory to evaluation of confidence in safety case," *Journal of Theoretical and Applied Information Technology*, vol. 47, no. 1, pp. 184–189, 2013.

[330] O. T. S. Jaradat and I. Bate, "Using safety contracts to guide the maintenance of systems and safety cases," in *Proceedings - 2017 13th European Dependable Computing Conference, EDCC 2017*, pp. 95–102, Institute of Electrical and Electronics Engineers Inc., 2017.

[331] O. Jaradat and S. Punnekkat, "Using safety contracts to verify design assumptions during runtime," in *Reliable Software Technologies–Ada-Europe 2018: 23rd Ada-Europe International Conference on Reliable Software Technologies, Lisbon, Portugal, June 18-22, 2018, Proceedings 23*, pp. 3–18, Springer, 2018.

[332] O. Jaradat, I. Bate, and S. Punnekkat, "Using sensitivity analysis to facilitate the maintenance of safety cases," in *Reliable Software Technologies–Ada-Europe 2015: 20th Ada-Europe International Conference on Reliable Software Technologies, Madrid Spain, June 22-26, 2015, Proceedings 20*, pp. 162–176, Springer, 2015.

[333] J. Brain, "Visual representation of safety cases," *Measurement and Control*, vol. 45, no. 3, pp. 82–86, 2012.

[334] J. Bragg and I. Habli, "What is acceptably safe for reinforcement learning?," in *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerras, Sweden, September 18, 2018, Proceedings 37*, pp. 418–430, Springer, 2018.

[335] X. Zhao, D. Zhang, M. Lu, and F. Zeng, "A new approach to assessment of confidence in assurance cases," in *Computer Safety, Reliability, and Security: SAFECOMP 2012 Workshops: Sassur, ASCoMS, DESEC4LCCI, ERCIM/EWICS, IWDE, Magdeburg, Germany, September 25-28, 2012. Proceedings 31*, pp. 79–91, Springer, 2012.