
Proceedings Paper:
Shahbeigi Roudposhti, Sepeedeh and Hawkins, Richard David (2025) Specifying Safety Requirements for Machine Learning Components in Autonomous Systems: A Survey. In: Safety Critical Systems Symposium (SSS '25).

Specifying Safety Requirements for Machine Learning Components in Autonomous Systems: A Survey

Sepeedeh Shahbeigi, Richard Hawkins

University of York

York, UK

Abstract Machine learning (ML) components are recognized for their potential to undertake tasks such as object detection and classification across a range of safety related applications. In order to be used safely, it is crucial that safety requirements for the ML components are correctly understood, specified in a manner that supports ML development, and can be demonstrated to be sufficient and valid. Traditional safety requirements approaches may not apply well to ML, due to their data-driven nature especially in complex environments. Defining safety requirements for ML components will require an understanding of the unique mechanisms by which ML can contribute to system safety and the potential failure modes of ML components. So far, little work has been done that attempts to systematically derive safety requirements specific to ML components and to ensure a traceable link between system-level and component-level safety requirements.This work aims to address this gap by providing a comprehensive survey of existing literature on methods for the elicitation of safety requirements for ML components. We identify key challenges and limitations in current methods and propose possible solutions. This paper highlights these issues and reviews current research to lay a foundation for developing robust and effective safety requirements for ML components.

1 Introduction

The integration of machine learning (ML) into autonomous systems (AS) represents a major technological advancement, leading to widespread adoption across diverse fields. In some of these applications, the undesirable behaviour of AS might result in harm to human lives or damage valuable property. Therefore, one of the most significant challenges for companies that develop safety-critical systems (SCS) is to assure (ensure and demonstrate) the safety of these systems (McDermid

et al., 2024). This is particularly challenging where AS operates in a complex environment and requires the creation of a compelling and comprehensive safety case.

One of the most critical and challenging steps in developing a safety case is to formulate a comprehensive, accurate, unambiguous, and testable set of safety requirements (SR). These requirements must be specified across multiple levels of autonomous system (AS) design decomposition, ensuring that they preserve the original intent and maintain traceability to the SRs from preceding levels (Hawkins et al., 2021). The literature on SCS has reported on many cases where systems have failed due to a lack in SRs specifications, or misunderstandings traced to problems in requirements engineering, contributing to accidents that cause damage to the environment, injury to people and even the loss of lives (Martins and Gorschek, 2016). One of the most notable example of this failure occurred with Uber's autonomous vehicle (AV) in March 2018 where the vehicle struck and killed a pedestrian in Tempe, Arizona.

Historically, software safety requirements have been derived through structured and standards- driven processes designed to mitigate risks using predictable and quantifiable methods. These approaches focus on ensuring that the software performs reliably under various conditions and gracefully fails when faults occur (Ebert, 2013). These structured approaches are governed by well-established standards that outline methodologies for deriving, specifying, and validating software SRs. Although these standards and practices provide a solid foundation for understanding and extending modern methodologies, they are insufficient to ensure the safety of AS, especially when they include ML components (Habiba et al., 2024).

Various methodologies have been developed to address the challenges of assuring AS, of which one of the most thorough is Safety Assurance of Autonomous Systems in Complex Environments (SACE) (Hawkins et al., 2021). SACE systematically integrates safety assurance of the AS into its development and also generates evidence for explicitly justifying the acceptable safety of AS.

SACE is a highly iterative eight stage process that runs in parallel with a baseline systems engineering and safety assurance process to modify, enhance, or add activities to address the challenges of safety assurance that comes with operation of an AS, Figure 1.

The SACE process defines a set of AS-specific safety assurance activities that supports the generation of artefacts that can be used as safety assurance evidence in a comprehensive safety case for an AS. In the first stage, SACE defines and validates the operating context for the AS, ensuring a clear understanding of the conditions under which the system will function. This validated context then informs stage two, where hazardous scenarios are identified and validated. These are situations that the AS might encounter during its operation that could lead to unsafe outcomes under certain conditions. Then in stage three, SACE focuses on defining and validating the safe operating concept (SOC), which includes specifying system SRs.

Stage four is particularly critical and highly relevant to this paper, as it focuses on defining and managing SRs across various levels of system decomposition. At each

tier, these requirements must accurately capture the intent established at the previous level. This process involves evaluating the proposed design for each tier to ensure that SRs are appropriately decomposed and allocated to relevant components. Accurate interpretation of these requirements for each component is essential, ensuring alignment with its specific design. Demonstrating this alignment goes beyond simply establishing traceable relationships between requirements at different tiers; it

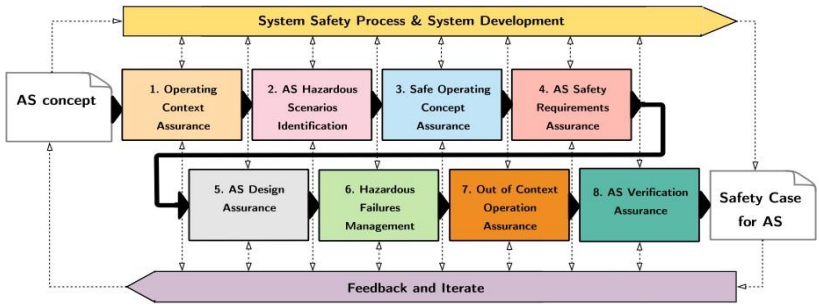


Fig.1: SACE process flow.

necessitates a clear explanation and justification of their adequacy and sufficiency, as illustrated in Figure 2.

Stage five focuses on developing design proposals to meet the SRs from Stage four. This involves iterative analysis, refinement, and updates to both the design and the requirements to ensure a robust and assured system across multiple tiers. Stages six, seven, and eight are responsible for identifying and mitigating hazardous failures, ensuring safe operation when out of defined context, and conducting thorough verification respectively. SACE does not cover the development of requirements for individual system components or their implementation. These aspects are addressed in other associated guidance documents. For example, the safety assurance of components implemented using ML is handled within the Assurance of Machine Learning for Autonomous Systems (AMLAS) framework (Hawkins et al., 2021).

AMLAS provides a six-stage process that describes the safety assurance activities that should be undertaken when developing an ML component for use as part of an AS. It also describes the artefacts that are generated from these activities and describes how these can be used to create a safety case for the ML component. While AMLAS offers valuable guidance on identifying ML-specific safety requirements and clarifying what needs to be demonstrated, it does not provide detailed methodologies or techniques for implementing these steps. Several studies have applied AMLAS to various use cases, demonstrating its practical applicability (Feather et al., 2022; Laher et al., 2022; Borg et al., 2023). Figure 3 illustrates the development lifecycle of an AS, highlighting the roles of SACE and AMLAS within this framework. This illustrates how SACE can be used to identify safety requirements for

components of an AS, and how AMLAS can then be used for any ML components to refine, implement and assure those safety requirements in an ML model. Ensuring a sufficient translation from component safety requirements defined from the system safety process using SACE to specific ML safety and data requirements using AMLAS is crucial to the overall safety of the AS. Traditional safety requirements processes will not be sufficient when considering ML components. Neither SACE nor AMLAS provide detailed guidance on how to define and structure clear and effective ML safety requirements, the criteria for assessing ML safety requirements or

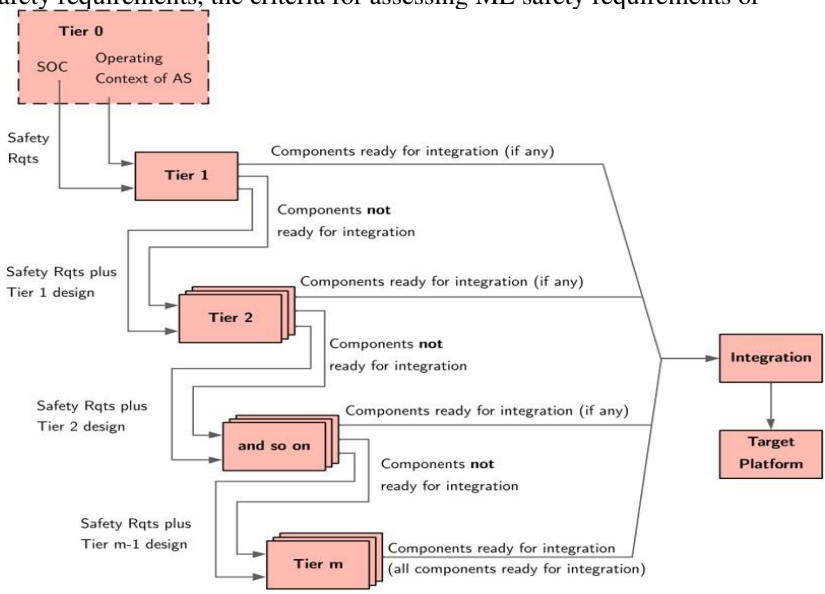
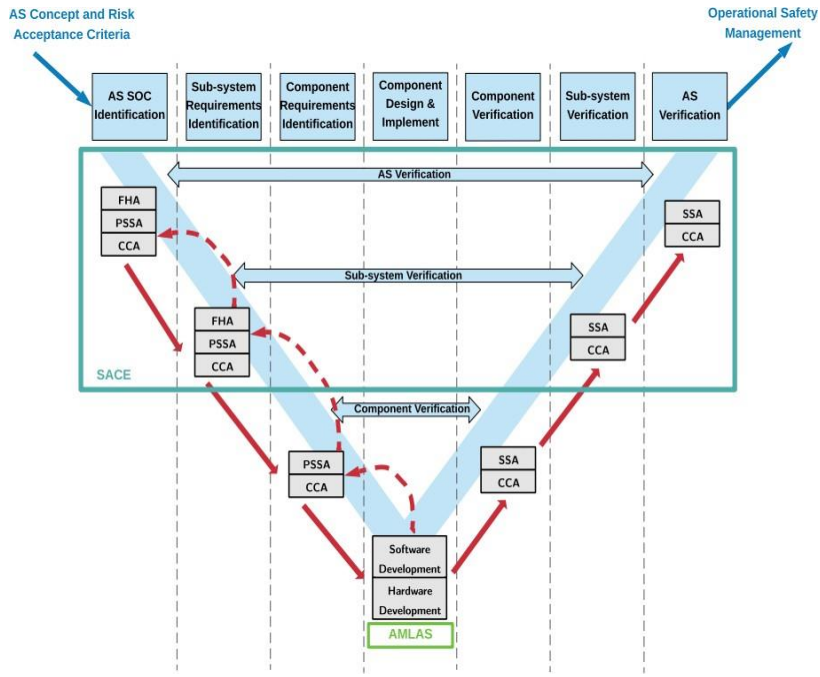


Fig. 2: The decomposition of AS safety requirements.

methods for demonstrating their traceability. This is a significant gap, as poorly defined or ambiguous requirements can undermine the overall safety and reliability of the AS (Kuwajima et al., 2020; Habiba et al., 2024).

This paper aims to address key research questions surrounding requirements engineering for ML in safety-critical systems:

- How does Requirements Engineering for ML differ from traditional requirements engineering approaches?
- What is a safety requirement Specification for ML?
- What has been done in the field of requirements for ML components?
- What are the existing gaps and challenges in developing SRs for ML-based systems?



The next section begins with an introduction to the concept of requirements for ML components, highlighting their significance in ensuring safe performance. This is followed by a discussion of key challenges in ML requirements, including specification, variability, traceability, data requirements, the emergence of new

Fig. 3: The scope of SACE and AMLAS.

requirements, and safety metrics. Each challenge is explained in detail, with a review of proposed solutions from the literature. The chapter concludes with a discussion and future directions, where we provide our perspective on these challenges from a safety standpoint. Finally, the paper summarizes the key insights and conclusions drawn from this analysis.

2 Requirements for ML component

A software requirement engineering process encompasses activities for eliciting, analysing, specifying, and validating software requirements, along with the management and documentation of requirements throughout the software product lifecycle (Belani et al., 2019). Introducing ML components into systems complicates the requirement development process compared to traditional components that follow a set

of predefined requirements (Vogelsang and Borg, 2019). Kuwajima et al. underscore the importance of defining clear requirements for ML components, as the absence of well-specified requirements can compromise the overall quality of these systems (Kuwajima et al., 2020). Additionally, the roles and responsibilities involved in requirement engineering (RE) processes for ML have evolved. For example, ML development's reliance on data highlights the importance of domain expertise throughout the ML lifecycle, as domain experts play a critical role in ensuring the relevance and applicability of the training data, model objectives, and interpretability of results (Von Rueden et al., 2021). Despite these transformations, the literature on RE for ML components mostly focuses on utilising AI to manage RE (Habiba et al., 2024). Furthermore, most of the research that focused on RE for ML does not consider it as part of a broader system.

The ML lifecycle typically comprises several iterative stages that address unique challenges distinct from traditional software development (Ashmore et al., 2021). The lifecycle starts with a data management stage, which includes data collection and preparation, where relevant data is sourced, cleaned, and labelled to ensure quality for model training. Following this stage is the model learning step, which involves selecting and tuning algorithms through experimentation, necessitating collaboration among data scientists to optimise performance. Once models are developed, they undergo rigorous evaluation based on both quantitative metrics and qualitative aspects such as fairness and interpretability. After successful evaluation, models are deployed, integrating them into broader systems while optimising for runtime performance and scalability. Finally, the workflow includes ongoing monitoring and maintenance, which are essential for ensuring that models maintain their performance in real-world conditions and can adapt to changes in data patterns (Amershi et al., 2019).

A fundamental activity in this lifecycle is to specify the requirements for the ML component. Requirement development is regarded as one of the most challenging tasks in ML development, as noted in several studies (Vogelsang and Borg, 2019; Perez-Cerrolaza et al., 2024; Ishikawa and Yoshioka, 2019). In this section, we review the literature on RE for ML, focusing on two key questions: What constitutes these requirements, and what challenges prevent the use of traditional methods? In this work, we are concerned with the requirements for the ML component, i.e. the model and data. The requirements for the development process are not the concern of this study. In the rest of this section, we will review the literature on the challenges of RE for ML and the proposed solutions to overcome these obstacles.

2.1 Specification

Specifying all expected behaviour of ML components may be impractical and unrealistic (Pei et al., 2022; Salay and Czarnecki, 2019). However, overlooking this step

completely might result in higher costs and complications as demonstrated by two examples, one in automotive and one in healthcare, in (Giunchiglia et al., 2023). Traditional software systems use fixed definitions for concepts, ensuring consistency across the system's functions. However, ML models lack a standardised approach for interpreting concepts; instead, they rely on datasets for training, which can introduce ambiguity and inconsistency in how concepts are recognised. This becomes especially evident in tasks such as perception. For example, a pedestrian detection function in an autonomous vehicle needs to detect pedestrians in their surroundings. However, "pedestrian" is an inherently unspecifiable concept, which leads to challenges in safety and assurance activities (Rahimi et al., 2019).

The ambiguity caused by the uncertain concept specification also affects the quality of the data used to train the ML. The supervised ML algorithms must generalise from training data to identify objects. Data labelling is labour-intensive and costly. labellers annotate the dataset based on their own concept definition, familiarity with the item being labelled. Based on the experiment conducted in (Kulesza et al., 2019), people labelling a set of web pages twice with a four-week gap between labelling sessions were, on average, only 81% consistent with their initial labels. This inconsistency in labelled data significantly contributes to the inconsistencies in input-output patterns which is difficult to detect.

To address the specification problem, Salay et al. propose a partial specification approach to bridge the safety assurance gap of developing ML components (Salay and Czarnecki, 2019). They discuss methods through which partial specification can be incorporated into the various stages of the ML development pipeline. Additionally, they discuss some safety related properties for specification language such as formality, uncertainty accommodation, and appropriateness. For instance, in the automotive domain, an effective specification language should account for the input domain's characteristics. Representing concepts like pedestrians using pixel information from camera images, such as defining them through kinematic wireframes, can be both intuitive and practical. This approach has proven successful in pedestrian detection methods such as histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), channel features or deformable part models (Cao et al., 2021).

To address the concept specification issue in the data labelling for ML training, which is referred to as concept evolution, Kulesza et al. propose a structured labelling tool that assists the labellers and was proven to improve the consistency of the annotation task (Kulesza et al., 2019).

A more recent approach for integrating domain knowledge into ML development is neuro- symbolic AI (Besold et al., 2021; Gaur et al., 2022; Hitzler et al., 2022). This method combines the symbolic reasoning of rule-based systems with the pattern-recognition capabilities of neural networks, allowing structured knowledge to complement data-driven learning [9]. In the context of RE for ML, neuro-symbolic AI ensures that domain-specific rules and constraints are consistently applied throughout the development process. This combination enhances both the robustness and interpretability of ML systems, as illustrated by Giunchiglia et al., who

demonstrate its effectiveness in applying domain knowledge across all stages of the ML lifecycle (Giunchiglia et al., 2023).

In conclusion, while fully specifying ML components remains impractical due to their inherently non-deterministic nature, partial specification presents a viable path forward. Defining what constitutes a sufficient partial specification is challenging, as it requires balancing formalised constraints with the flexibility to accommodate uncertainty and evolving concepts. Runtime verification and monitoring will be essential in bridging the gaps left by partial specifications, ensuring that ML components continue to meet safety requirements under dynamic conditions. Incorporating domain knowledge into the specification process and enforcing its consistency across development can benefit from modern approaches like neuro-symbolic AI, which integrates structured rules with data-driven learning to enhance robustness and reliability. Table 1 summarises the keypoints from the literature for the specification challenges for ML.

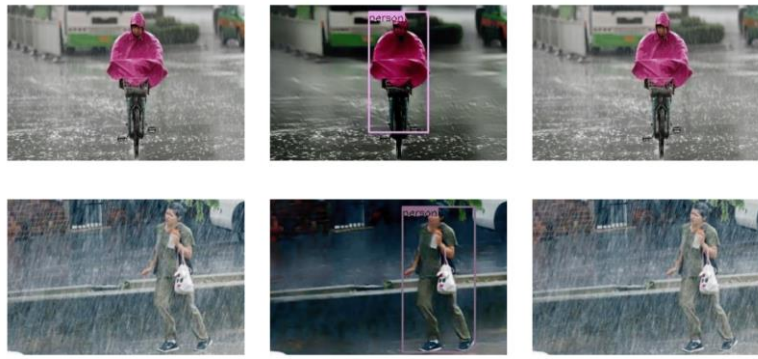
Table 1: Key Challenges and Solutions for ML Component Specification

Challenge	Description	Proposed Solutions
Incomplete Concept Specification	Specifying all behaviours is impractical due to ML's non- deterministic nature (Pei et al., 2022; Salay and Czarnecki, 2019).	Partial specifications: define key requirements with flexibility to accommodate uncertainty (Salay and Czarnecki, 2019).
Ambiguity in Concept Recognition	ML models lack standard definitions, leading to inconsistencies in concept recognition (Rahimi et al., 2019).	Use intuitive representations (e.g., kinematic wireframes) for perception task (Dalal and Triggs, 2005; Cao et al., 2021).
Inconsistent Data Labelling	Labellers' subjective interpretations lead to inconsistent annotations (Kulesza et al., 2019).	Structured labelling tools improve consistency and definition coherence (Kulesza et al., 2019).
Concept Evolution in Training Data	Changing definitions of concepts affects the reliability of training data (Kulesza et al., 2019).	Continuous monitoring and adaptive processes to maintain robustness.
Lack of Domain Knowledge Integration	Traditional ML lacks structured integration of domain knowledge (Besold et al., 2021; Gaur et al., 2022; Hitzler et al., 2022).	Neuro-symbolic AI integrates symbolic reasoning with neural networks to apply domain knowledge consistently (Giunchiglia et al.).

2.2 Variability

ML components must generalise from training data to identify objects under diverse and sometimes unpredictable conditions. Such variability necessitates performance robustness requirements (Hawkins et al., 2021). For example, a trained model must

detect a pedestrian in rainy weather conditions as well as a sunny one, Fig. 4. While robustness is a critical requirement for ML systems, a formal definition remains lacking (Drenkow et al., 2021; Freiesleben and Grote, 2023). There may be various reasons and deployment scenarios where robustness is essential, each driving attention toward different aspects of this property. This, in turn, leads to a focus on distinct sub-types of robustness.



(**a**) Input (**b**) With proposed (**c**) Without proposed

Figure 4: Machine learning algorithms likelihood of failure in presence of adverse weather condition increases unless a mechanism is utilised to reduce the noise due to these conditions. (a) input images, (b) Enhanced images using algorithm proposed by (Y. Liu, 2021), and (c) the YOLOv4 ML algorithm fails to detect the pedestrian in these two images.

For instance, in computer vision tasks, robustness against image noise—such as Gaussian noise, occlusions, or adversarial perturbations—is particularly crucial (França et al., 2021). In contrast, for natural language processing (NLP) systems, robustness might involve handling noisy inputs like typos, slang, or ambiguous phrasing (Wang et al., 2021).

Additionally, it must be noted that the same failure modes due to lack of robustness might have different criticalities depending on the operating context. For instance, missing a speed limit sign on an empty highway poses less risk than in a busy one, illustrating how situational factors influence requirement prioritisation and specification. Reflecting this need, Kuwajima et al. argue that attributes tied to operational scenarios, such as time of day, weather, and road type, should be incorporated into the requirements for ML (Kuwajima et al., 2020). Furthermore, a task-aware risk assessment of the perception failure of an AV would be helpful in understanding and establishing the relationship between the failures in perception and those in navigation (Antonante et al., 2023). In another work, researchers attempt to specify and test robustness against geometric transformations requirements by benchmarking it against human perception in autonomous driving applications (Hu et al., 2020). They argue that the robustness performance of the ML should match that of a human

driver, namely, if a human driver can detect an object under a transformation so should an ML component.

To capture these situational factors systematically, the automotive industry relies on the concept of the operational design domain (ODD). Integrating requirements with ODD has given rise to new challenges in RE for automotive applications, such as determining the appropriate level of detail for ODD features and establishing a standardised representation. Furthermore, functionalities that detect entry into and exit from different ODDs are now required (Habibullah et al., 2024). Given this dependence on the operational context, RE for ML must adapt to complex and often unpredictable system behaviours. This adaptation necessitates requirements that are continuously validated and refined throughout the lifecycle to account for changes in data, evolving model behaviour, and shifting operational contexts (Amershi et al., 2019; Belani et al., 2019).

Table 2: Challenges and Solutions for Variability in ML behaviour

Challenge	Description	Proposed Solutions
Variability in Environmental Conditions	ML components must generalise to diverse, unpredictable conditions (e.g., weather, time of day) (Hawkins et al., 2021).	Define robustness requirements that address variations in operational conditions like weather, lighting, and road type (Kuwajima et al., 2020).
Lack of Formal definition for Robustness	No formal, unified definition for robustness in ML systems across domains (e.g., computer vision vs NLP) (Drenkow et al., 2021; Freiesleben and Grote, 2023).	Establish specific robustness subtypes for various applications, such as image noise resilience in vision and typo handling in NLP (França et al., 2021; Wang et al., 2021).
Context-Sensitive Failure Modes	The criticality of failure modes varies by operating context (e.g., failure in detecting a speed limit sign) (Hawkins et al., 2021).	Incorporate operational context features, like time, weather, and traffic conditions, into requirement specification and prioritisation (Kuwajima et al., 2020).
Operational Design Domain (ODD)	Defining the operational context for ML systems is complex and lacks standardisation, particularly in automotive applications (Habibullah et al., 2024).	Develop standardised representation of ODD features and functionalities for detecting transitions between different ODDs (Habibullah et al., 2024).
Dynamic and Evolving Operational Contexts	ML components must adapt to changes in data, model be- haviour, and operational con- text over time (Amershi et al., 2019; Belani et al., 2019).	Implement an iterative process for continuous validation and refinement of requirements to accommodate dynamic operational conditions (Amershiet al., 2019).

2.3 Traceability

Another issue in RE for systems with ML components is traceability. Unlike traditional software systems where traceability links between requirements, design, and implementation can be well-defined, ML-based systems add complexity due to their black box treatment (Pei et al., 2022). Since ML models learn from data and continuously update their responses, establishing direct traceability between initial requirements and system outputs becomes challenging. Traceability becomes even more elusive as ML models are frequently re-trained and refined, creating a disconnection between requirements and the evolving model behaviour over time (Belani et al., 2019).

However, despite the importance of this issue, the literature reveals a lack of concrete solutions for achieving effective traceability in systems with ML components. Instead, the focus is mostly on using ML models to manage the requirements. Still even these proposed solutions do not seem practical for industrial adoption (Maro et al., 2018). Husen et al. propose a top-down methodology consisting of six different modelling techniques including KAOS goal model, UML, STAMP/STPA, and safety case analysis (Husen et al., 2022). They then use a matrix to demonstrate the related elements between these models. However, they do not discuss the rationale behind the relationships or how these models apply. Additionally, Aravantinos et al. (Aravantinos and Diehl, 2018) propose replacing conventional low-level requirements (LLRs) with deep-neural-network-specific artifacts such as training datasets, neural network architectures, and learning configurations. They also propose a domain coverage model, similar to ODD concept, to ensure dataset relevancy. However, further detail is necessary to evaluate the applicability of their work (Aravantinos and Diehl, 2018). A summary of the traceability challenges and solution in ML-based systems is provided is Table 3.

Table 3: Challenges and Proposed Solutions for Traceability in ML Systems

Challenge	Description	Proposed Solutions
Complex Traceability	Traditional traceability links between requirements, design, and implementation are difficult to maintain in ML systems due to their evolving, data-driven nature (Pei et al., 2022; Belani et al., 2019).	Limited practical solutions: focus has been on managing requirements with ML models (Maro et al., 2018). Rich traceability with satisfaction arguments might address this (Dick, 2002).
Lack of Standardised Methods	Continuous model updates and retraining causes disconnections between initial requirements and	Husen et al. propose a top-down methodology with six modelling techniques (e.g., KAOS, UML,

	outputs, com- plicating traceability (Belani et al., 2019).	STAMP/STPA) and matrix- based traceability, but without rationale for relationships (Husen et al., 2022).
Evolving Model Behav- iour	ML systems' dynamic behaviour makes it difficult to trace back to original requirements over time (Pei et al., 2022).	Aravantinos et al. suggest using DNN-specific artefacts (e.g., training datasets, architectures) instead of low-level requirements, along with a domain coverage model for dataset relevancy (Aravantinos and Diehl, 2018). Further evaluation needed.

2.4 Data requirements

Since there is a shift for ML components from hard-coded rules to learning-enabled models, there is now a new requirement on the data. There have been many works that explore this new type of requirement. For example, Hawkins et al. suggest that the requirement on data can be categorised into relevancy, completeness, accuracy, and balance criteria (Hawkins et al., 2021). Kuwajima et al. state the ISO/IEC 25000 series mention of system and software quality requirements and evaluation (SQuaRE), which defines a set of quality characteristics for product, data, in use, and IT service models (Kuwajima et al., 2020). The data quality model consists of characteristics such as accuracy, completeness, consistency, currency, and understandability. Gauerhof et al. provide a pedestrian detection example to demonstrate how the desiderata (desired properties), defined in (Ashmore et al., 2021), can be used to define a set of safety requirements on the data used for ML development (Gauerhof et al., 2020). To further consider the collaborative nature of data requirement development, Dey et al. introduce a three-layer framework with a verifiable template for eliciting data requirements (Dey and Lee, 2023). Jahić et al. propose a structured, domain specific language called SEMKIS-DL that focuses on linking customer requirements directly to dataset construction and neural network (NN) performance metric (Jahić et al., 2023).

While many researchers highlight properties such as correctness, sufficiency, and consistency as important properties of datasets, these terms are subjective and call for a reference for comparison. In that regard, Barzamini et al. propose a framework which develops a benchmark to evaluate the dataset requirements (Barzamini and Rahimi, 2022). To the best of our knowledge, this is the only existing work, that we found, that looks into this challenge. Their framework, illustrated in Figure 4, searches and vets relevant concepts and their variations in a specific domain utilising methods such as NLP. On the other hand, the same concepts are identified in the dataset. Comparing the coverage of domain concepts specifications and the dataset

one, the level of discrepancy between the two determines the quality of the dataset. A summary of the challenges and solutions related to the data requirement for ML-based systems in this section is presented in Table 4.

Table 4: Challenges and Proposed Solutions for Data Requirements for ML Systems

Requirement Type	Description	Proposed Solutions
Data Quality	New requirements focus on data relevancy, completeness, accuracy, and balance due to ML's data-driven nature (Hawkins et al., 2021).	ISO/IEC 25000 SQuaRE framework outlines quality criteria like accuracy, completeness, consistency, and understandability (Kuwajima et al., 2020).
Safety and Relevancy	Ensuring datasets meet safety requirements are crucial for ML applications like pedestrian detection (Ashmore et al., 2021).	Gauerhof et al. use desiderata to define safety-related data requirements for ML development (Gauerhof et al., 2020).
Collaborative Requirement Development	Eliciting data requirements involves collaboration among stakeholders to ensure relevance and completeness (Dey and Lee, 2023).	Dey et al. propose a three-layer framework with a verifiable template for collaborative data requirement development (Dey and Lee, 2023).
Domain-Specific Linkage	Linking requirements to dataset construction and NN performance is challenging due to subjectivity (Jahić et al., 2023).	SEMKIS-DL, a structured language, links customer requirements directly to datasets and performance metrics (Jahić et al., 2023).
Benchmarking and Evaluation	Terms like correctness, sufficiency, and consistency need objective benchmarks for comparison (Barzamini and Rahimi, 2022).	Barzamini et al. propose a framework to develop benchmarks for evaluating dataset requirements (Barzamini and Rahimi, 2022).

2.5 Emergence of new requirements

The need for transparency in ML decisions has introduced new requirement categories focused on interpretability and explainability (Pei et al., 2022; Habiba et al., 2024). In safety-critical applications, for instance, stakeholders must understand and trust the system's decisions, requiring RE processes to specify not only functional requirements but also interpretability standards that make model behaviour traceable and explainable (Rudin, 2019; Doshi-Velez and Kim, 2017). Methods such as model

visualisation or employing interpretable surrogate models are being explored to help fulfil these requirements, allowing stakeholders to interpret complex ML behaviour without compromising performance (Arrieta et al., 2020).

The integration of ML has also expanded RE's interdisciplinary scope, requiring inputs from fields such as ethics, law, and data security to address concerns around bias, fairness, and privacy (Mittelstadt et al., 2016; Raji and Buolamwini, 2019; Lepri et al., 2018; Koopman and Wagner, 2017). As ML systems increasingly influence areas with societal impacts, capturing requirements that reflect ethical and legal considerations is essential for ensuring fairness and protecting user rights (Holstein et al., 2019).

2.6 Safety metrics

To evaluate the performance of machine learning models, it is essential to understand the fundamental concepts of prediction outcomes. A True Positive (TP) occurs when the model correctly predicts a positive instance. A True Negative (TN) is when the model correctly predicts a negative instance. A False Positive (FP), also known as a Type I error, happens when the model incorrectly predicts a positive instance. Conversely, a False Negative (FN), or Type II error, occurs when the model incorrectly predicts a negative instance. Using these notions, we can define the most commonly used performance metrics (Powers, 2020):

- Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$ represents the overall correctness of the model by measuring the proportion of correct predictions (both positives and negatives) out of all predictions.
- Precision $= \frac{TP}{TP+FP}$ measures the proportion of correctly predicted positive instances among all instances predicted as positive. It indicates the model's ability to avoid false positives.
- Recall $= \frac{TP}{TP+FN}$ Also known as sensitivity, recall quantifies the proportion of actual positives correctly identified by the model. It is particularly important in safety-critical applications where missing a positive instance (false negative) could have severe consequences.
- F1 score $= 2 \times \frac{precision \times recall}{precision + recall}$ Provides a balanced measure by taking the harmonic mean of precision and recall. It considers both false positives and false negatives, making it useful for evaluating performance on imbalanced datasets.

Although certain metrics, like recall, can provide partial indications of safety performance, they exhibit significant limitations in safety-critical applications. For instance, as demonstrated by Berry et al. (Berry, 2022), a high or low recall does not

directly translate into safe or unsafe ML operation. The relationship between these metrics and safety is complex, influenced by factors such as the operational context and the quality of the model. This complexity makes developing appropriate safety metrics challenging and often application specific. To address these shortcomings and incorporate uncertainty into performance metrics for a more conservative safety assessment, Herd et al. (Herd and Burton, 2024) propose considering factors such as sample size, model calibration, and dataset coverage into these performance metrics.

4 Discussions and future directions

The specification challenges associated with ML components in AS are multifaceted and deeply interconnected. The emergence of data-driven models introduces unique complexities that differ significantly from traditional software systems, necessitating a shift in RE approaches. These challenges, such as incomplete concept specification, ambiguity in data labelling, robustness requirements, and the lack of effective traceability, are not isolated but form a complex web where each issue impacts and is influenced by others. For instance, incomplete or inconsistent concept specification directly affects data labelling quality, leading to ambiguous training datasets that compromise model performance. This ambiguity exacerbates robustness issues, as models trained on inconsistent data may fail under diverse or unforeseen conditions. Evolving operational contexts, such as changing weather conditions or varying road types (for AVs), further complicate the situation by necessitating continuous updates to datasets and model parameters, making traceability even more challenging. Addressing any single challenge in isolation is unlikely to yield significant improvements; a holistic approach that considers these interdependencies is essential.

Traceability, a cornerstone of traditional RE, is particularly complex in ML-based systems due to their dynamic and data-driven nature. Establishing clear links between requirements, design, implementation, and verification becomes challenging when models learn from data and continuously evolve. Rich traceability frameworks, specifically those incorporating satisfaction arguments, offer a promising solution to this issue. Satisfaction arguments provide a structured rationale explaining how specific requirements are met through the system's design and operation (Dick, 2002). By extending this concept to ML, traceability can encompass not only static links between artifacts but also dynamic relationships involving data, model behaviour, and contextual factors. For example, linking performance metrics to specific data requirements and operational scenarios can help ensure that the system remains aligned with its intended purpose, even as the model evolves. This approach can also aid in validating partial specifications by demonstrating that critical safety and performance criteria are consistently satisfied across different stages of development and deployment.

Despite these potential solutions, many challenges in specifying and ensuring the safety of ML components remain unresolved. Partial specification methods and data quality evaluation frameworks provide valuable tools, but they do not fully address the inherent uncertainties and evolving nature of ML systems. Robustness requirements, for example, still lack a formal, universally accepted definition, and existing approaches often focus on specific aspects, such as image noise or adversarial attacks, without considering the broader operational context. Moreover, practical solutions for effective traceability are still in their infancy. While proposals such as structured labelling tools and domain- specific languages offer partial solutions, they have yet to see widespread adoption in industry. Integrating these tools into existing workflows and ensuring their scalability remain significant challenges.

5 Conclusion

In conclusion, specifying safety requirements for ML components in AS presents a complex and evolving challenge. Unlike traditional software, ML systems introduce new dimensions of uncertainty, particularly in areas such as incomplete concept specification, data ambiguity, variability, and traceability. While partial specifications, structured labelling tools, and emerging frameworks like neuro-symbolic AI offer promising approaches, significant gaps remain. Effective solutions require a holistic, multidisciplinary approach that addresses the interconnections between these challenges, and considers evolving contextual factors. Continued research and collaboration are essential to develop practical, scalable methods that ensure the safety of ML-based systems in real-world applications.
To explicitly address the research questions outlined in the introduction:

- **How does Requirements Engineering for ML differ from traditional requirements engineering approaches?**

RE for ML differs from traditional approaches due to the dynamic, data-driven, and iterative nature of ML systems. Unlike conventional systems with fixed, predefined requirements, ML systems rely on training datasets to generalise concepts, leading to inherent ambiguities and inconsistencies in recognising and interpreting tasks like perception. Traceability is also more complex, as linking requirements to implementation is challenging due to the evolving nature of ML models. Additionally, RE for ML requires the integration of domain-specific knowledge, ethical considerations, and adaptability to shifting operational contexts, which are less prominent in traditional frameworks. Furthermore, ML development involves iterative processes, including data preparation, model training, and continuous evaluation, demanding ongoing validation and refinement of requirements throughout the system's lifecycle. These differences underscore the inadequacy of traditional methods and the need for tailored RE approaches to address ML's unique challenges.

- **What is a safety requirement specification for ML?**

A safety requirement specification for ML includes not only traditional functional requirements but also specific requirements tailored to the unique aspects of ML systems. These include data requirements like relevance, completeness, accuracy, and balance, as well as robustness requirements to handle variability in environmental conditions and adversarial scenarios. Safety requirement specifications must also address traceability by linking high-level system goals to ML-specific components, ensuring alignment between design and operational outcomes. Additionally, interpretability and fairness are critical, as ML systems must provide explanations for their behaviour and ensure unbiased outcomes across diverse user groups. Tools like partial specification approaches, structured labelling techniques, and neuro-symbolic AI have been proposed to support the development of such specifications, enabling the integration of domain knowledge and enhancing the robustness of ML components in safety-critical contexts.

- **What has been done in the field of requirements for ML components?**

Our survey of the state-of-the-art show significant advancements in methodologies such as partial specifications, use of neuro-symbolic AI, and structured labelling tools. These methods address issues like concept evolution and the integration of domain knowledge, providing a foundation for more robust requirements for ML components.

- **What are the existing gaps and challenges in developing SRs for ML-based systems?**

Existing gaps and challenges in developing safety requirements (SRs) for machine learning (ML)-based systems include the lack of formalised and universally accepted robustness definitions and effective traceability frameworks. Robustness requirements are often narrowly defined, addressing specific aspects like image noise or adversarial attacks, without considering the broader operational context of ML components. Ambiguity in data labelling and the inconsistent specification of concepts further exacerbate these challenges, as ML models rely heavily on datasets for training, which may introduce variability and inconsistency in recognising key elements. Additionally, existing frameworks like AMLAS and SACE provide valuable guidance but fail to offer detailed methodologies for structuring, assessing, and implementing SRs, leaving significant gaps in defining and demonstrating their adequacy and sufficiency. Practical tools, such as structured labelling or domain-specific languages, are emerging but remain underdeveloped and lack widespread adoption, further hindering the integration of SRs into established workflows.

5.1 References

S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann. Software engineering for machine learning: A case study. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pages 291–300. IEEE, 2019.

P. Antonante, S. Veer, K. Leung, X. Weng, L. Carlone, and M. Pavone. Task- aware risk estimation of perception failures for autonomous vehicles. arXiv preprint arXiv:2305.01870, 2023.

V. Aravantinos and F. Diehl. Traceability of deep neural networks. arXiv preprint arXiv:1812.06744, 2018.

A. B. Arrieta, N. D´ıaz-Rodr´ıguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado,
S. Garc´ıa, S. Gil-L´opez, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion, 58:82–115, 2020.

R. Ashmore, R. Calinescu, and C. Paterson. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. ACM Computing Surveys (CSUR), 54(5):1–39, 2021.

H. Barzamini and M. Rahimi. Cade: The missing benchmark in evaluating dataset requirements of ai-enabled software. In 2022 IEEE 30th International Requirements Engineering Conference (RE), pages 64–76. IEEE, 2022.

H. Belani, M. Vukovic, and Z˘. Car. Requirements engineering challenges in building ai- based complex systems. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pages 252–255. IEEE, 2019.

D. M. Berry. Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence? In International Working Conference on Requirements Engineering: Foundation for Software Quality, pages 19–25. Springer, 2022.

T. R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Ku¨hnberger, L. C. Lamb, P. M. V. Lima, L. de Penning, et al. Neural-symbolic learning and reasoning: A survey and interpretation 1. In Neuro-Symbolic Artificial Intelligence: The State of the Art, pages 1–51. IOS press, 2021.

M. Borg, J. Henriksson, K. Socha, O. Lennartsson, E. Sonnsj¨o L¨onegren, T. Bui,
P. Tomaszewski, S. R. Sathyamoorthy, S. Brink, and M. Helali Moghadam.

Ergo, smirk is safe: a safety case for a machine learning component in a pedestrian automatic emergency brake system. Software quality journal, 31(2):335–403, 2023.

J. Cao, Y. Pang, J. Xie, F. S. Khan, and L. Shao. From handcrafted to deep features for pedestrian detection: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(9):4913–4934, 2021.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

S. Dey and S.-W. Lee. A multi-layered collaborative framework for evidence-driven data requirements engineering for machine learning-based safety-critical systems. In Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, pages 1404–1413, 2023.

J. Dick. Rich traceability. In Proceedings of the 1st international workshop on traceability in emerging forms of software engineering, pages 18–23. Citeseer, 2002.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608, 2017.

N. Drenkow, N. Sani, I. Shpitser, and M. Unberath. A systematic review of robustness in deep learning for computer vision: Mind the gap? arXiv preprint arXiv:2112.00639, 2021.

C. Ebert. Functional safety industry best practices for introducing and using iso 26262. Technical report, SAE Technical Paper, 2013.

M. S. Feather, P. C. Slingerland, S. Guerrini, and M. Spolaor. Assurance guidance for machine learning in a safety-critical system. In 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pages 394–401. IEEE, 2022.

H. L. Fran¸ca, C. Teixeira, and N. Laranjeiro. Techniques for evaluating the robustness of deep learning systems: A preliminary review. In 2021 10th Latin-American Symposium on Dependable Computing (LADC), pages 1–5. IEEE, 2021.

T. Freiesleben and T. Grote. Beyond generalization: a theory of robustness in machine learning. Syntheses, 202(4):109, 2023.

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, pages 197–212. Springer, 2020.

M. Gaur, K. Gunaratna, S. Bhatt, and A. Sheth. Knowledge-infused learning: A sweet spot in neuro-symbolic ai. IEEE Internet Computing, 26(4):5–11, 2022.

E. Giunchiglia, F. Imrie, M. van´ader´aSchaar, and T. Lukasiewicz. Machine learning with requirements: A manifesto. Neurosymbolic Artificial Intelligence, (Preprint):1–13.

U.-e. Habiba, M. Haug, J. Bogner, and S. Wagner. How mature is requirements engineering for ai-based systems? a systematic mapping study on practices, challenges, and future research directions. Requirements Engineering, pages 1–34, 2024.

K. M. Habibullah, H.-M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss,

H. Sivencrona, and P. J. Li. Requirements and software engineering for automotive perception systems: an interview study. Requirements Engineering, pages 1–24, 2024.

R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli. Guidance on the assurance of machine learning in autonomous systems (amlas). arXiv preprint arXiv:2102.01564, 2021.

B. Herd and S. Burton. Can you trust your ml metrics? using subjective logic to determine the true contribution of ml metrics for safety. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, pages 1579–1586, 2024.

P. Hitzler, A. Eberhart, M. Ebrahimi, M. K. Sarker, and L. Zhou. Neuro-symbolic ap- proaches in artificial intelligence. National Science Review, 9(6): nwac035, 2022.

K. Holstein, J. Wortman Vaughan, H. Daum´e III, M. Dudik, and H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In Proceedings of the 2019 CHI conference on human factors in computing systems, pages 1–16, 2019.

B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. Selim, and M. Chechik. Towards requirements specification for machine-learned perception based on human performance. In 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), pages 48–51. IEEE, 2020.

J. H. Husen, H. Washizaki, H. T. Tun, N. Yoshioka, Y. Fukazawa, and H. Takeuchi. Trace- able business-to-safety analysis framework for safety-critical machine learning systems. In Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, pages 50–51, 2022.

F. Ishikawa and N. Yoshioka. How do engineers perceive difficulties in engineering of machine-learning systems? questionnaire survey. In 2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP), pages 2–9. IEEE, 2019.

B. Jahić, N. Guelfi, and B. Ries. Semkis-dsl: A domain-specific language to support requirements engineering of datasets and neural network recognition. Information, 14 (4):213, 2023.

P. Koopman and M. Wagner. Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intelligent Transportation Systems Magazine, 9(1):90–96, 2017.

T. Kulesza, D. Charles, R. Caruana, S. A. Amershi, and D. A. Fisher. Structured labeling to facilitate concept evolution in machine learning, June 11 2019. US Patent 10,318,572.

H. Kuwajima, H. Yasuoka, and T. Nakae. Engineering problems in machine learning systems. Machine Learning, 109(5):1103–1126, 2020.

S. Laher, C. Brackstone, S. Reis, A. Nguyen, S. White, and I. Habli. Review of the amlas methodology for application in healthcare. arXiv preprint arXiv:2209.00421, 2022.

B. Lepri, N. Oliver, E. Letouz´e, A. Pentland, and P. Vinck. Fair, transparent, and ac- countable algorithmic decision-making processes: The premise, the proposed solutions, and the open challenges. Philosophy & Technology, 31(4):611–627, 2018.

Liu, Y., Ma, J., Wang, Y., & Zong, C. (2020). A novel algorithm for detecting pedestrians on rainy image. *Sensors*, *21*(1), 112.

S. Maro, J.-P. Steghofer, and M. Staron. Software traceability in the automotive domain: Challenges and solutions. Journal of Systems and Software, 141:85–110, 2018.

L. E. G. Martins and T. Gorschek. Requirements engineering for safety-critical systems: A systematic literature review. Information and software technology, 75:71–89, 2016.

J. A. McDermid, R. Calinescu, I. Habli, R. Hawkins, Y. Jia, J. Molloy, M. Osborne,

C. Paterson, Z. Porter, and P. R. Conmy. The safety of autonomy: A systematic approach. Computer, 57(4):16–25, 2024.

B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi. The ethics of algorithms: Mapping the debate. Big Data & Society, 3(2):2053951716679679, 2016.

Z. Pei, L. Liu, C. Wang, and J. Wang. Requirements engineering for machine learning: A review and reflection. In 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW), pages 166–175. IEEE, 2022.

J. Perez-Cerrolaza, J. Abella, M. Borg, C. Donzella, J. Cerquides, F. J. Cazorla, C. Englund, M. Tauber, G. Nikolakopoulos, and J. L. Flores. Artificial intelligence for safety- critical systems in industrial and transportation domains: A survey. ACM Computing Surveys, 56(7):1–40, 2024.

D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061, 2020.

M. Rahimi, J. L. Guo, S. Kokaly, and M. Chechik. Toward requirements specification for machine-learned components. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pages 241–244. IEEE, 2019.

I. D. Raji and J. Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pages 429–435, 2019.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence, 1(5):206–215, 2019.

R. Salay and K. Czarnecki. Improving ml safety with partial specifications. In Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38, pages 288–300. Springer, 2019.

A. Vogelsang and M. Borg. Requirements engineering for machine learning: Perspectives from data scientists. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pages 245–251. IEEE, 2019.

L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Transactions on Knowledge and Data Engineering, 35(1):614–633, 2021.

X. Wang, H. Wang, and D. Yang. Measure and improve robustness in nlp models: A survey. arXiv preprint arXiv:2112.08313, 2021.

……..

Instructions for Completing the SCSC License to Publish Form

The form may be printed out and filled in manually, but please write clearly

Note when printing that either A4 or Letter size paper may be used. If your printer is single sided only, set so as just to print Page 2.

In all cases, the Title and Author(s) name(s) are required. If an organization holds the copyright, its name and postal address is also required.

When complete, and if not already done, print out a copy of the form for signature.

If the copyright owner is an organization, a representative must sign the form, thus:

<signature of representative

<name of representative>, on behalf of <name of organisation> <date of signature>

Alternatively, if the Author(s) retain(s) the copyright, each is to sign the form:

<signature of Author> <date of signature>

When the form is complete and signed, please scan it and send it by e-mail to the SCSC editor who sent you the form to complete, retaining the master for your records. If a scanner is not available, a sufficiently high-resolution photograph would suffice.

Title of Article: Specifying Safety Requirements for Machine Learning Components in Autonomous Systems: A survey

_____

Name of Author(s): Sepeedeh Shahbeigi, Richard Hawkins

_____

Name & Address of Copyright Owner (if not author):

_____

_____

1. Safety-Critical Systems Club and SCSC are trading names of the Safety-Critical Systems Club C.I.C. a Community Interest Company limited by guarantee and registered in England and Wales with company number 13084663. Registered Office: Southgate Chambers, 37/39 Southgate Street, Winchester, SO23 9EH.
2. By signing this form, you (the copyright owner or authorized representative of the owner):
   a. Agree to grant to the Safety-Critical Systems Club (the publisher) the non-exclusive right to publish, distribute or broadcast your paper in printed or digital form, e.g. in the SCSC Newsletter, the SCSC eJournal, SCSC books, and online through our website.

   b. Assert that the paper is your original work. If it contains material which is someone else's copyright, you assert that you have obtained the unrestricted permission of the copyright owner, and that the material is clearly identified and acknowledged in the text.
   c. Assert that the paper does not, to the best of your knowledge, contain anything which is libelous, illegal or infringes anyone's copyright or other rights, and that appropriate releases have been obtained from people identifiable in any images included.
   d. Assert your Moral Rights to be identified as the author, if appropriate.
3. We confirm that we will respect the rights of the author(s) and will make sure that their name(s) are always closely associated with the paper.
4. Copyright remains with the copyright owner(s) and we will acknowledge this. However, you authorize the Safety Critical Systems Club, if we wish, to act on your behalf to defend copyright.

Signatures:

Sepeedeh Shahbeigi
Richard Hawkins

.....

If you foresee any problem with entering into this agreement, please contact me without delay.

You should insert your own copyright details in the page 1 footnote of your paper; this will normally name you or your employer as the owner of the copyright.

Acknowledgments Any acknowledgments should appear at the end of the paper, before the reference list. It is usual to acknowledge significant contributors or reviewers.

Disclaimers Any disclaimers should be presented at the end in this format.

References

Alexander C, Ishikawa S, Silverstein M et al (1977) A pattern language: towns, buildings, construction. Oxford University Press

Armoush A, Salewski F, Kowalewski S (2009) Design pattern representation for safety-critical embedded systems. J Softw Eng Appl 2:1-12

Pierce R, Fowler D (2010) Applying IEC 61508 to air traffic management. In: Dale C, Anderson T (eds) Making systems safer. Springer

RAEng (2011) Global navigation space systems: reliance and vulnerabilities. Royal Academy of Engineering. http://www.raeng.org.uk/gnss. Accessed 31 August 2011