Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey

JON PEREZ-CERROLAZA, Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain
JAUME ABELLA, Barcelona Supercomputing Center (BSC), Spain
MARKUS BORG, RISE Research Institutes of Sweden AB, Sweden
CARLO DONZELLA, Exida, Italy
JESÚS CERQUIDES, Artificial Intelligence Research Institute (IIIA-CSIC), Spain
FRANCISCO J. CAZORLA, BSC and Maspatechnologies S.L., Spain
CRISTOFER ENGLUND, RISE Research Institutes of Sweden AB, Sweden
MARKUS TAUBER, Research Studios Austria, Austria
GEORGE NIKOLAKOPOULOS, Luleå University of Technology, Sweden
JOSE LUIS FLORES, Ikerlan Technology Research Centre, BRTA, Spain

Artificial Intelligence (AI) can enable the development of next-generation autonomous safety-critical systems in which Machine Learning (ML) algorithms learn optimized and safe solutions. AI can also support and assist human safety engineers in developing safety-critical systems. However, reconciling both cutting-edge and state-of-the-art AI technology with safety engineering processes and safety standards is an open challenge that must be addressed before AI can be fully embraced in safety-critical systems. Many works already address this challenge, resulting in a vast and fragmented literature. Focusing on the industrial and transportation domains, this survey structures and analyzes challenges, techniques, and methods for developing AI-based safety-critical systems, from traditional functional safety systems to autonomous systems. AI *trustworthiness* spans several dimensions, such as engineering, ethics and legal, and this survey focuses on the safety engineering dimension.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Computer systems organization** → **Dependable and fault-tolerant systems and networks**; **Robotics**; **Robotic autonomy**; • **Hardware** → **Safety critical systems**.

Additional Key Words and Phrases: functional safety, autonomous systems

Authors' addresses: Jon Perez-Cerrolaza, Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Arrasate / Mondragon, Spain, jmperez@ikerlan.es; Jaume Abella, Barcelona Supercomputing Center (BSC), Barcelona, Spain, jaume.abella@bsc.es; Markus Borg, RISE Research Institutes of Sweden AB, Lund, Sweden, markus.borg@ri.se; Carlo Donzella, Exida, Rovereto, Italy, carlo.donzella@exida-dev.com; Jesús Cerquides, Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain, cerquide@iiia.csic.es; Francisco J. Cazorla, BSC and Maspatechnologies S.L., Barcelona, Spain, francisco.cazorla@bsc.es; Cristofer Englund, RISE Research Institutes of Sweden AB, Gothenburg, Sweden, cristofer.englund@ri.se; Markus Tauber, Research Studios Austria, Vienna, Austria, markus.tauber@researchstudio.at; George Nikolakopoulos, Luleå University of Technology, Luleå, Sweden, george.nikolakopoulos@ltu.se; Jose Luis Flores, Ikerlan Technology Research Centre, BRTA, Arrasate / Mondragon, Spain, jlflores@ikerlan.es.

0360-0300/2023/10-ART $15.00
https://doi.org/10.1145/3626314

1 INTRODUCTION

Artificial Intelligence (AI) is at the core of recent scientific and industrial advances, such as Autonomous Driving (AD) [Chen et al. 2021; Grigorescu et al. 2020; Kiran et al. 2021] and Unmanned Aerial Vehicles (UAVs) [Liu et al. 2020; Torens et al. 2022]. AI technology is a cross-domain innovation driver for numerous novel application use cases [ISO 2021] and embedded intelligence-driven solutions [Jenn et al. 2020; Serpanos et al. 2020]. In some specific high-integrity application scenarios, AI is increasingly "used to support safety-critical decisions where errors can lead to catastrophic and fatal consequences" [Castelvecchi 2016; ISO 2021b; Perez et al. 2021] (e.g., AD [Koopman and Wagner 2016; Riedmaier et al. 2020; Salay et al. 2018], railway interlocking [Athavale et al. 2020a; Klein 1991; Nordland 2004; Perez et al. 2021], aircraft collision avoidance [Julian et al. 2019], UAVs [Dill et al. [n. d.]; Sarathy et al. 2019; Schirmer et al. [n. d.]; Torens et al. 2022]).

In this line, it is acknowledged that AI is "one of the only technically and economically viable" technologies for developing autonomous systems [Jenn et al. 2020]. Driven by AD and UAV engineering challenges and the associated economic investment, there is a significant research and engineering effort to define novel technical solutions for developing AI-based autonomous systems [Grigorescu et al. 2020; Hand and Khan 2020; Koopman and Wagner 2016; Rajabli et al. 2021; Riedmaier et al. 2020; Salay et al. 2018], neaten with the updating and definition of novel safety standards [ARP 2023; ISO 2019; ISO 2021b; ULSE 2020] to deal with AI-specific traits. These solutions are also of interest for multiple transportation domains such as avionics [Athavale et al. 2020b; Harrison et al. 1993], railway [Athavale et al. 2020b,a; Nordland 2004; Perez et al. 2021] and automotive [Salay and Czarnecki 2018; Tabani et al. 2019], and industrial domain applications such as robotics [Täubig et al. 2012] and driverless industrial trucks [ISO 2020a, 2021]. In all of these domains, AI technologies can be used to develop both traditional functional safety systems, as well as next-generation autonomous safety-critical systems [Berghoff et al. 2020; ISO 2021b; Jenn et al. 2020; VDE 2021].

However, existing AI software technologies have several generic limitations related to compliance with current safety standards [Berghoff et al. 2020; Jenn et al. 2020]. The most notorious include the 'black box' nature of AI solutions causing limitations regarding their explainability and analyzability [Ackerman 2017; Castelvecchi 2016; Guidotti et al. 2018; Salay et al. 2018; Torens et al. 2022; Ward and Habli 2020], and compliance limitations concerning software development lifecycle phases, such as specification correctness and completeness, design, testing, verification and validation [Hand and Khan 2020; Koopman and Wagner 2016; Mainzer 2020; Menzies and Pecheur 2005; Nordland 2004; Pereira and Thomas 2020; Rajabli et al. 2021; Torens et al. 2022; Vassev 2016]. Due to these limitations (challenges), AI techniques have not been recommended for use in safety-critical systems [CENELEC. 2020; IEC 2010; Nordland 2004]. In fact, nowadays, there are still no structured development approaches, methods and tools with generic acceptance for developing AI-based safety-critical systems [Berghoff et al. 2020; Putzer et al. 2021]. The evolving normative landscape also attests to this with the recent AI [ARP 2023; CENELEC 2020; ISO 2021b], Safety Of The Intended Functionality (SOTIF) [ISO 2019] and autonomous systems safety standards [ULSE 2020; VDE 2021] that are in development (drafts) or recently published with limited consolidation of industry best practices [Berghoff et al. 2020; Feth et al. 2018; Jenn et al. 2020].

These complexities are compounded by a significant fragmentation of the research contributions targeting the use of AI for developing autonomous systems with [Tiusanen et al. 2020] and without specific safety considerations [Mainzer 2020], different safety AI challenges [Amodei et al. 2016; He et al. [n. d.]; Jenn et al. 2020], multiple use cases [Athavale et al. 2020b; ISO 2021], multiple types of AI [Feldt et al. 2018], different lifecycle phases (e.g., design [Perez et al. 2021; Varadaraju 2007], test [Čegiň 2020; Hourani et al. 2019; Huang et al. 2020], verification [Akella et al. 2020; Ehlers 2017; Huang et al. 2017]), generic AI solutions (e.g., reinforcement learning [Arulkumaran et al. 2017]) and safety adaptations (e.g., safe reinforcement learning [García and Fernández 2015]), with references to multiple existing [IEC 2010; ISO 2018] and novel domain-specific safety standards [Chemweno et al. 2020; ISO 2019; Putzer et al. 2021; Tiusanen et al. 2020; ULSE 2020; VDE 2021].

Trust becomes paramount in paving the way for the industrial development, commercialization and societal adoption of AI-based safety-critical systems such as AD systems [Widen and Koopman 2022] and UAVs [Torens et al. 2022]. AI *trustworthiness* spans several dimensions, such as engineering, ethics and legal, and this survey focuses on the safety engineering dimension. This survey provides an overview and categorization of the vast and fragmented research contributions that target the development of AI-based safety-critical systems for industrial and transportation domains, from traditional Functional Safety (FuSa) to autonomous safety-critical systems. This survey targets researchers and safety engineers concerned with the diligent development of AI-based safety-critical systems in a context where the technology novelty leads to a lack of consolidated industry best practices, and available safety standards have little or no consideration for AI technology [Erben et al. 2006].

Figure 1 provides a graphical representation of the survey structure in which we categorize and summarize selected key research contributions toward using AI technology for (i) the development of AI-based safety-critical systems (*product*) in Section 4, (ii) runtime learning/adaptation of AI-based safety-critical systems (*runtime*) in Section 5, and (iii) the development *process* of safety-critical systems in Section 6. Previous Sections 2 and 3 describe the basic concepts, terminology and taxonomy used in the remainder of this work. Section 7 discusses *trustworthiness* as a multidimensional (e.g., engineering, ethics, legal) and multidisciplinary foundation for developing and adopting AI-based safety-critical systems. Lastly, Section 8 summarizes the overall conclusion and outlines future research directions.

<table>
<tr><th><b>Product</b> (§4)</th><th><b>Runtime</b> (§5)</th><th><b>Process</b> (§6)</th></tr>
<tr><td>AI System (§4.1)
AI Item (§4.2)</td><td>Runtime Learning/Adapt. (§5.1)</td><td>AI Safety Engineering (§6.2)</td></tr>
<tr><td colspan="2">Execution Platform (Inference) (§4.3)
Tools and Training Platform (§4.4)</td><td>Traditional Safety Eng. (§6.1)</td></tr>
<tr><th colspan="3"><b>Trustworthiness</b> (§7)</th></tr>
<tr><td colspan="3">Engineering Dimension (§7.1)
Ethical Dimension (§7.2)
Legal Dimension (§7.3)</td></tr>
</table>

Fig. 1. Diagram summarizing the structure of this survey

2 BACKGROUND

We next summarize basic concepts and terms used in the survey like AI (§2.1), FuSa standards (§2.2) and ML properties (§2.3). This survey uses existing dependable and secure computing terminology [Avižienis et al. 2004], the AI terminology defined in ISO 22989 [ISO 2021], and the FuSa terminology defined by safety standards IEC 61508-4 [IEC 2010] and ISO 26262-1 [ISO 2018]. This survey also integrates terminology from various research fields as described in the referenced survey publications.

2.1 Artificial Intelligence (AI)

As stated in the VDE-AR-E 2842-61 standard, “there is no generally accepted definition of artificial intelligence” [VDE 2021]. Furthermore, Feldt et al. [Feldt et al. 2018] claim that “there is not even a consensus around what AI is” (referring to the scope of types of algorithms and models). Nonetheless, ISO 22989 provides an ’engineering system' oriented definition of AI used in this survey [ISO 2021]: "set of methods or automated entities that together build, optimize and apply a model so that the system can, for a given set of predefined tasks, compute predictions, recommendations, or decisions".

The term *AI safety* [Amodei et al. 2016; Everitt et al. 2018] is commonly used in the literature to describe techniques and methods that aim to avoid or mitigate the potential harm that developed AI technology applications could produce to humanity. However, within this survey, the term *AI safety* refers to AI-related techniques, processes, and methods that aim to comply with applicable safety standards (§2.2, §3.3). Thus, this is a narrower and more focused definition.

Finally, Machine Learning (ML) is "the art and science of letting computers learn without being explicitly programmed" [Henriksson et al. 2018]. It is a subfield of AI that uses algorithms to learn from example training data sets that implicitly specify the intended functionalities, features, rules and constraints. The learning process can be, for instance, *supervised* (using labelled data), *unsupervised* (not using labelled data), *semisupervised* (using both labelled and unlabelled data) and *reinforcement learning* ("a machine learning agent(s) learns through an iterative process by trial and error") [Arulkumaran et al. 2017; García and Fernández 2015; ISO 2021]. When the learned ML solution executes on an embedded system (electronics/software implementation with *model parameters*), it performs inferences in which the ML solution provides online actionable outputs based on the inputs provided. Finally, the generic statement that most of the contributions labeled as AI are in fact ML contributions [Jordan 2019] is also extensible to the research contributions analyzed in the scope of the given survey.

2.2 Functional Safety (FuSa) Standards

The development of safety-critical systems follows stringent certification or assessment processes in accordance with generic and domain-specific safety standards defined by national and international standardization organizations (e.g., ISO) and associations (e.g., Verband Deutscher Elektrotechniker (VDE)). FuSa is defined as "part of the overall safety" of a system that assures the "freedom from unacceptable risk" [IEC 2010], through safety functions embedded in programmable electronics systems (electronics/software). IEC 61508 [IEC 2010] is a reference generic FuSa standard for industrial (e.g., industrial machinery [ISO 2015], robotics [ISO 2011], tractors, machinery for agriculture [ISO 2018c]) and ground transportation domains (automotive [ISO 2018], railway [CENELEC. 2020]). Notably, FuSa standards from the air transportation domain (e.g., avionics [RTCA 2011; SAE 2010], space [Pelton and Jakhu 2010]) "do not consider IEC 61508 as a reference safety standard" [Perez-Cerrolaza et al. 2022]. Yet, they also focus on risk mitigation due to failures in safety functions embedded in programmable electronic systems. Further information concerning FuSa standards and associated certification or assessment processes can be found elsewhere [Machrouh et al. 2012; Martinez et al. 2018].

Among all FuSa standards, there is significant variability in terms, definitions and requirements. For example, IEC 61508 defines the Safety Integrity Level (SIL) with a range of discrete values from lowest to highest integrity (SIL1 - SIL4). And the equivalent in the automotive industry is Automotive Safety Integrity Level (ASIL) (ASILA - ASILD) and in avionics Design Assurance Level (DAL) (DAL E - DAL A). In this survey, we use the generic IEC 61508 as the reference safety standard and take into technical consideration the ground transportation and industrial domains listed above. We also use automotive ISO 26262, given that automotive AD challenges have attracted a significant number of research publications.

For the most critical systems (SIL4, DAL A), "the probability of a dangerous failure is in the range of 10^{-9} per hour of operation, that is, approximately one dangerous failure every 114.155 years" [Perez-Cerrolaza et al. 2022]. Thus, the associated error rate is multiple orders of magnitude smaller than the error rate considered excellent for generic AI solutions (e.g., 99% accuracy) [Koopman et al. 2021]. Attaining such an extremely low probability of dangerous failures requires handling *systematic errors* (e.g., human error, tool error) and *random errors* (e.g., memory bit flip) according to strict safety methods, processes, and techniques. FuSa standards are denoted in the survey as traditional because the first versions were defined decades ago, and the referenced techniques and methods are based on best practices consolidated in the industry over the last decades. Nonetheless, FuSa standards are also updated to accommodate novel and evolving technologies (e.g., ISO 26262-11 for semiconductors technology).

2.3 ML Properties

Due to the intrinsic stochastic nature of ML training and associated epistemic uncertainties [Varshney 2016; VDE 2021], the achievable confidence usually depends on "complex hypotheses" [Jenn et al. 2020] related to the different properties of the training and inference input data (e.g., data drift, distribution, correlation), their coverage (e.g., edge/corner cases, hidden variable) and metrics [VDE 2021]. In this vein, the safety argumentation of systematic errors management is commonly based on high-level AI-related properties adapted to the context of safety systems [Jenn et al. 2020]. For example, as defined by [Jenn et al. 2020]:

- *Auditability*: "Extent to which an independent examination of the development and verification process of the system can be performed".
- *Data Quality*: "Extent to which data are free of defects and possess desired features".
- *Explainability/Interpretability*: "Extent to which a ML system can provide an explanation about a decision in a form understandable by a human" (e.g., see surveys [Adadi and Berrada 2018; Barredo Arrieta et al. 2020; Guidotti et al. 2018]).
- *Monitorability*: "Extent to which a system provides information that allows to discriminate a *correct behavior* from an *incorrect behavior*".
- *Provability*: "Extent to which mathematical guarantees can be provided that some functional or non-functional properties are satisfied" (e.g., formal verification).
- *Robustness*: "Ability of the system to perform its intended function in the presence of: a) Abnormal inputs (e.g., sensor failure), b) Unknown inputs (e.g., unspecified conditions)".

Nonetheless, several research initiatives aim to mitigate this stochastic nature and simplify the safety argument by enforcing deterministic training processes [Nagarajan et al. 2019]. Furthermore, the ML model implementation can be either deterministic (e.g., a Neural Network (NN) produces the same outputs given the same inputs [VDE 2021]) or stochastic [Cummings and Bauchwitz 2022] if the implementation includes techniques that rely on internal random variables.

3 TAXONOMY

This section summarizes the taxonomy used in the survey to classify Types of AI (TAIs) (§3.1), levels of automation (§3.2), heteronomous and autonomous safety standards (§3.3), point of application of AI technology (§3.4), and AI safety engineering (§3.5). This taxonomy aims to provide neutral classification criteria and definitions of terms, reconciling the high variability of terms and concepts from research contributions and safety standards. For instance, the proposed taxonomy can potentially map to domain-specific terms and concepts such as VDE-AR-E2842-61 standard terms [VDE 2021], e.g., *AI-based system* ('system level'), *AI item* ('AI element'), *AI safety engineering* ('AI-blueprint').

3.1 Type of AI (TAI)

There is a lack of consensus about TAIs in the research community [Feldt et al. 2018; Jordan 2019]. Some works propose as a starting point the 'five tribes of AI' [Domingos 2018], on which this section builds on and adds optimization algorithms to classify the TAIs used in referenced research publications within the survey scope.

(1) *Connectionists* are design learning algorithms based on optimization techniques such as gradient descent, where models are represented as Neural Networks (NNs) and specialized Deep Learning (DL) models [Grigorescu et al. 2020; Pouyanfar et al. 2018] such as Deep Neural Networks (DNNs) [Liu et al. 2017], Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and autoencoders.
(2) *Bayesians* are probabilistic outcome-based graphical model representations for probabilistic inference such as Bayesian and Markov networks.
(3) *Symbolists* are logic-focused algorithms such as rule-based programming (e.g., "always stop in front of a stop sign"), Constraint Programming (CP), decision trees (e.g., random decision forest [Törnblom and Nadjm-Tehrani 2018]), fuzzy logic [Kurd and Kelly 2004] and rational agents [Kamali et al. 2017].
(4) *Analogizers* are similarity-based classification algorithms (e.g., Support Vector Machine (SVM)).
(5) *Optimization* algorithms aim to discover optimum or satisfactory solutions performing iterative updates and comparison procedures (e.g., Genetic Algorithm (GA)).

And as summarized in Table 1 and the white paper on auditable AI systems [Berghoff et al. 2020], *connectionist* is the most common TAI embedded in safety-critical systems (*product*, *runtime*), and it is commonly used in the development *process* (e.g., DL-driven test scenario generation for DL-based *products*).

Table 1. Types of AI (TAIs) per point of application analyzed in the survey

Type of AI (TAI)	**Point of Application (PA)**		
	Product	***Runtime***	***Process***
Analogizers	-	-	[Daramola et al. 2013]
Bayesians	-	-	[Akella et al. 2020; Fan et al. 2020; Gal 2016; Gangopadhyay et al. 2019; Hutter et al. 2019; Jesenski et al. 2019; Jha et al. 2019; Kendall et al. 2015; Kendall and Cipolla 2016; Simon et al. 2019; Wang and Zhao 2018]
Connectionits	[Ackerman 2017; Al-Khoury 2017; Al-Sharman et al. 2021; Beglerovic et al. 2018; Borg et al. 2018; Bosio et al. 2019; Chen et al. 2021; Corsi et al. 2020; Ehlers 2017; Geißler et al. 2021; Grigorescu et al. 2020; Hains et al. 2018; Harel-Canada et al. 2020; Henriksson et al. 2018; Huang et al. 2020; Jacobsson 2005; Jäger et al. 2018; Julian et al. 2019, 2016; Katz et al. 2017; Kendall et al. 2015; Kendall and Cipolla 2016; Kiran et al. 2021; Kurd et al. 2007; Li et al. 2017; Lisboa 2001; Liu et al. 2017; O'Brien et al. 2020; Parisi et al. 2019; Pouyanfar et al. 2018; Pulina and Tacchella 2012; Pullum et al. 2007; Rahman et al. 2021; Ruospo et al. 2020; Schuman et al. 2017; Schumann and Liu 2010; Sun et al. 2019a; Tabani et al. 2020; Taylor 2006; Taylor et al. 2003b; Yurii and Liudmila 2017; Zhang and Li 2020]	[Johnson et al. 2001; Kurd and Kelly 2005; Osborne et al. 2021; Taylor et al. 2003b]	[Beglerovic et al. 2018; Čegiň and Rástočný 2020; Daramola et al. 2013; Jenkins et al. 2018; Krajewski et al. 2018]
Optimization	[Klein 1991; Theuretzbacher 1987]	[Trojaola et al. 2020]	[El-Serafy et al. 2015; Gheraibia et al. 2018; Perez et al. 2021; Tuncali et al. 2020]
Symbolists	[Kamali et al. 2017; Törnblom and Nadjm-Tehrani 2018]	[Kurd and Kelly 2004, 2005]	[Bagschik et al. 2018; Daramola et al. 2013; Godboley et al. 2021; Jacobsson 2005, 2006; Kruber et al. 2019; Kurd and Kelly 2004; Li et al. 2020; Ouazraoui and Nait-Said 2019; Sallak et al. 2006; Waymo 2019]

3.2 Autonomous, Heteronomous, Automation, Automatic and Collaborative Systems

There is a high diversity of taxonomies to classify autonomous systems and levels of automation, from generic taxonomies [Frohm 2008; ISO 2021; Kugele et al. 2021; Sheridan and Verplank 1978] to domain-specific taxonomies such as automotive AD [SAE 2014], avionics [Clough 2002; EASA 2021], railway [IEC 2009, 2014] and robotics

[Beer et al. 2014; Guiochet et al. 2017; SPARC 2016]. Hence, as for the AI term definition, there is a lack of cross-domain definition consensus for these terms. However, ISO 22989 [ISO 2021] provides basic generic definitions adaptable to the scope of the survey:

- *Autonomous* systems operate in an 'open environment' (e.g., AD systems operate in an "open parameter space in which an infinite number of different traffic situations can occur" [Riedmaier et al. 2020]) without human-in-the-loop control and supervision (e.g., AD SAE level 5 [SAE 2014], avionics 3B [EASA 2021], generic levels 7-10 [Sheridan and Verplank 1978]). As defined by ISO 22989, *autonomy* constitutes the highest level of automation in which "the system is capable of modifying its operating domain or its goals without external intervention, control or oversight" [ISO 2021].
- The term *heteronomous* system [ISO 2021] encompasses different levels of automation that must operate in a '(semi-)open environment' with varying degrees of human collaboration, control and supervision, and integrates the generic term 'semi-autonomous'. For example, AD SAE levels 1-4 [Ma et al. 2020; SAE 2014], avionics levels 1A-1B-2-3A [EASA 2021], railway systems Grade of automation (GoA) 1-4 [IEC 2009, 2014] and generic levels 2-6 [Sheridan and Verplank 1978]. *Automation/automated* is defined as "pertaining to a process or system that, under specified conditions, functions without human intervention" [ISO 2021].
- *Automatic* systems operate in a 'closed environment' with well-defined safety rules and constraints known at design time [Guiochet et al. 2017]. Thus, the system is neither *autonomous* nor *heteronomous*. It simply executes an automation of safety functions without human intervention (e.g., railway interlocking system [Klein 1991]) in compliance with applicable FuSa standards.
- *Collaborative* robot refers to diverse robot-human collaborative working models ranging from *automatic* (e.g., safety-rated monitored stop) to *heteronomous* and *autonomous* working models [ISO 2016; Rodríguez-Guerra et al. 2021], and combinations of the previous.

3.3 Heteronomous and Autonomous Safety Standards

Table 2 classifies the most relevant FuSa, heteronomous and autonomous safety standards (draft standards are represented in parentheses and standards that explicitly consider AI technology are underlined), and identifies among the dozens of AI standardization initiatives [CENELEC 2020] those that target the development of AI-based safety-critical systems. The recommended 'reading map' for AI practitioners/professionals not specialized in safety-critical systems is the reading of generic and automotive domain FuSa (IEC 61508; ISO 26262), heteronomous/autonomous (VDE-AR-E2842-61; ISO/PAS 21448, UL 4600), and AI standards for safety systems (ISO 5469; ISO/AWI PAS 8800).

3.3.1 Heteronomous Safety Standards. The development of novel types of safety-related systems, such as Advanced Driver-Assistance Systems (ADAS) [Mainzer 2020], led to a novel scenario where safety-critical systems could fail even in the absence of an electronic/software failure. For example, the intended safety function fails due to unexpected operating conditions not considered in the perception ML algorithm training [Koopman et al. 2019]. Thus, there was a need for a novel type of safety standards, complementary with FuSa standards, such as the automotive domain SOTIF [ISO 2019]. For example, the development of an ML algorithm-based safety perception function integrated into a safety ADAS, requires compliance with the associated SOTIF (e.g., ISO/PAS 21448), applicable AI standards (e.g., ISO 5469, ISO/AWI PAS 8800), and the embedded implementation should comply with the associated FuSa standard (e.g., ISO 26262). Some transportation and industrial domains have already defined domain-specific safety standard drafts [Rodríguez-Guerra et al. 2021; Sarathy et al. 2019; Tiusanen et al. 2020] (e.g., automotive SAE levels 3-4 [ISO 2020b, 2021a]; mining and earth moving machinery [ISO 2017, 2018b], autoguidance systems for tractors and machinery for agriculture [ISO 2009], highly automated agricultural machines [ISO 2018a], collaborative robots [ISO 2016], aircraft systems with complex functions [ASTM 2021]). And some of these standards do not mention or consider AI, as they could potentially be implemented with

Table 2. Summary of selected FuSa, AI, heteronomous and autonomous safety-critical systems standards

Domains		Safety Standards			AI standards for safety systems	Reviews / Surveys
		FuSa	Heteronomous	Autonomous		
Transp.	Space	ECSS-Q-ST-30C/40C	-		-	[Machrouh et al. 2012; Martinez et al. 2018]
	Railway	EN 5012x	IEC 62290, IEC 62267		-	[Machrouh et al. 2012; Martinez et al. 2018; Tiusanen et al. 2020]
	Avionics	ARP4754, DO-178C	ASTM F3269-21		(ARP6983)	[Machrouh et al. 2012; Martinez et al. 2018][Torens et al. 2022]
	Automotive	ISO 26262	ISO/PAS 21448	ISO 4804, ISO 5083, (UL 4600)	(ISO/AWI PAS 8800)	[Koopman et al. 2019; Machrouh et al. 2012; Martinez et al. 2018]
Industrial	Robotics	ISO 10218-1	-		-	[Rodríguez-Guerra et al. 2021]
	Mining & earth moving machinery	EN ISO 19014	ISO 17757, ISO 16001, ISO 18758-2		-	-
	Ind. Machinery	ISO 13849-1	(ISO/TR 22100-5), (ISO 3691-4)		-	[Anastasi et al. 2021]
	Agriculture	ISO 25119	ISO 10975, ISO 18497		-	-
Generic		IEC 61508	VDE-AR-E2842-61		(ISO 5469)	[Machrouh et al. 2012; Martinez et al. 2018]

different technologies. For example, the machinery domain ISO 22100 technical report [ISO 2021] describes risk reduction approaches for driverless industrial trucks implemented with or without AI technology. But, within the scope of the survey, we only consider the scenarios where the system is developed with AI technology.

3.3.2 Autonomous Safety Standards. The development of autonomous safety systems leads to a novel scenario in which the safety system makes autonomous decisions without human control/supervision in an open environment. For these novel types of safety systems, which can not be developed and certified with previously described standards (only), the automotive industry has defined several specific standards, such as UL 4600 [ULSE 2020]. Regarding industrial domains, some authors provide an overview and review of industrial safety standards [Tiusanen et al. 2020], such as autonomous machine systems [ISO 2019] and driverless industrial trucks [ISO 2020a].

Finally, the VDE-AR-E2842-61 [VDE 2021] ("development of trustworthiness of autonomous/cognitive systems") is a generic standard (draft) for developing Autonomous/Cognitive (AC) systems. This standard combines *SOTIF*, *heteronomous* and *autonomous* system considerations with AI technology.

3.4 Point of Application, Usage Level (UL) and Class

This Section briefly reconciles research [Feldt et al. 2018] and ISO 5469 standard taxonomies [ISO 2021b] concerning the AI technology usage type, class and characteristics. The *point of application* taxonomy proposed by Feldt et al. [Feldt et al. 2018] defines both 'when' and 'on what' an AI technology is applied using three categories that can be adapted to the survey scope as follows.

(1) *Product*: A safety-critical system (the *product*) relies on offline embedded AI technology to perform one or more safety functions. As summarized in Figure 3, the AI-based safety-critical system is composed of one or more *AI-based systems* that integrate one or more *AI items*. The *AI item* embeds the AI technology in an electronic/software component [EASA 2021] with required model parameters, and it is deployed and executed on a given *execution platform* (e.g., GPU).

(2) *Runtime*: The AI-based safety-critical system integrates AI technology with runtime field learning capability (online). A *runtime* can also be considered a *product* variant that integrates *dynamic reconfiguration* (IEC 61508-7 C.3.10) and becomes a 'one of a kind' system.
(3) *Process*: AI technology can support and facilitate the offline development of a safety function (*safety engineering*) in compliance with the techniques, methods and processes required by applicable safety standards. This is applied during the system development process, but the used AI technology itself is not embedded into the system (unlike a *product*/*runtime*).

The ISO 5469 *Usage Level (UL)* taxonomy [ISO 2021b; Schneider 2021] classifies the use of AI technology using four basic levels (*A-D*) that can be related to the previously described *point of application* taxonomy. In a *product*/*runtime*, a safety function can be implemented using AI technology (*A*), or a non-safety-related function that could interfere with safety function(s) (*C*) or be interference-free (*D*). Furthermore, AI technology can also be used in the safety-critical development *process* (*B*). UL *A* and *B* are further classified based on whether the AI performs automated decisions (*A1, B1*) or not (*A2, B2*). Based on this, AI-based diagnostic functions can be classified as *A2* or *C*. And, as a rule of thumb, the UL of AI-items performing autonomous safety functions is *A1*, while AI-items for automatic, heteronomous and collaborative safety-critical systems may be *A1* or *A2*.

Finally, the ISO 5469 *class* I-II-III taxonomy [ISO 2021b; Schneider 2021] defines whether a given AI technology can be used for the development of a given safety-critical system (*product*/*runtime*/*process*) in compliance with previously described safety standards (see §2.2, §3.3). *Class I* solutions can be developed and reviewed in compliance with safety standards (e.g., use of formal verification [Perez et al. 2021]). *Class II* solutions cannot be developed and reviewed in compliance with safety standards, but the proposed compensation measures are sufficient for that purpose. For example, the *safety bag*/*diverse monitor* (C.3.4 IEC 61508-7) technique (a.k.a., run-time checker), safely monitors that the results provided by an AI item are safe [IEC 2010; Theuretzbacher 1987]. So, the *safety bag* becomes the safety function that prevents unsafe states, and the AI item does not require safety standard compliance. Finally, *Class III* solutions cannot be developed and reviewed in compliance with safety standards, and compensation measures are insufficient. For example, AI-based ADAS using class III AI technology are not considered safety-critical systems, and the driver itself is responsible for driving the vehicle, monitoring the ADAS operation and taking vehicle control in a short time if the ADAS detects and notifies that can no longer provide the intended functionality [Cummings and Bauchwitz 2022; Koopman et al. 2021; Widen and Koopman 2022]. And if sufficient compensation measures are defined (e.g., human expert verification, safety bag) a *Class III* solution becomes a *Class II* solution.

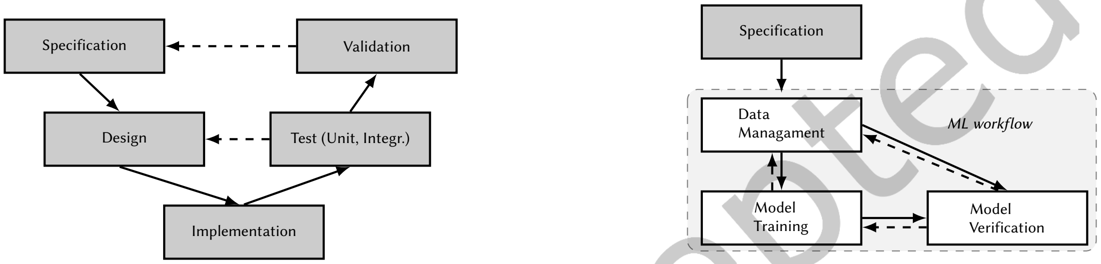
3.5 Traditional Safety Engineering and AI Safety Engineering

The *traditional safety engineering* of a safety-critical system follows a V-model development lifecycle as mandated by safety standards (e.g.,'realization' phase IEC 61508 [IEC 2010], 'product development' ISO 26262 [ISO 2018]) with the following generic phases (see Figure 2a): specification, design, implementation, Verification, Validation and Testing (VVT). The verification activity must confirm that the result of all the development phases (i.e. specification, design, test, and validation) meets the assigned objectives and safety development requirements (IEC 61508-4 §3.8.1). And the validation activity must confirm by examination of the evidence (e.g., test results) that the specification has been met (IEC 61508-4 §3.8.2) [IEC 2010].

VDE-AR-E2842-61-1 [VDE 2021] states that *AI technology* should be considered a third type of technology (in addition to electronics and software) due to its unique characteristics (e.g., uncertainty-related failures). Thus, *AI safety engineering* refers to the engineering lifecycle, processes, activities and techniques required to develop AI-based (sub)systems and AI *items* [Putzer et al. 2021]. The ISO 5469 [ISO 2021b] standard defines a high-level lifecycle that combines the V-model and ML lifecycle activities. Furthermore, the VDE-AR-E2842-61-5 [VDE 2021] standard states that different TAIs might require different processes and lifecycles (still to be defined).

For example, while some optimization-based solutions can be developed using a V-model approach [Klein 1991; Perez et al. 2021], most of the analyzed research contributions use a ML workflow [Ashmore et al. 2021; Rabe et al. 2021] or hybrids [Kuwajima et al. 2020]. Also, Rabe et al. provide an automotive domain specific survey of ML development methodologies [Rabe et al. 2021]. In any case, a relevant difference between *traditional safety engineering* and ML workflows is that the former is specification-driven and the latter data-driven [Rabe et al. 2021].

Figure 2b shows the simplified ML lifecycle based on Ashmore et al. [Ashmore et al. 2021] used in the survey that, starting from a system *specification* phase [Bencomo et al. 2022], follows a ML workflow with *data management*, *model learning* and *model verification* phases. The resulting verified model is then deployed to an execution platform. And the model execution can feed the data management phase with operational data for future model releases.



(a) Simplified V-model

(b) Simplified ML lifecycle [Ashmore et al. 2021]

Fig. 2. Simplified lifecyle for *traditional safety engineering* (V-model) and *AI safety engineering* for ML

4 PRODUCT - AI-BASED SAFETY-CRITICAL SYSTEM

This section describes the challenges, techniques, and methods used to develop AI-based safety-critical systems (the *product*) from traditional FuSa to autonomous systems. The description structure follows the *product* layers presented in Section 3.4 and summarized in Figure 3: *AI system* (§4.1), *AI item* (§4.2), and inference *execution platform* (§4.3). We also provide a brief summary of tools and training platforms (§4.4).

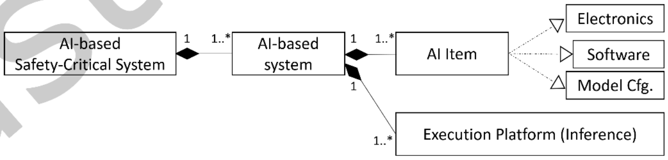


Fig. 3. *Product* composition diagram (UML)

Table 3 summarizes selected AI safety techniques for the development of AI-based safety-critical *products*. AI technology (*Class I-II*) has already been used for the development of specific FuSa compliant *automatic* safety-critical systems (e.g., SIL4 railway interlocking [Klein 1991]). Basically, there are two basic approaches for the development of AI-based FuSa systems: the safety verification of all possible input and output combinations either offline using formal verification (*class I*) [Vassev 2016] or online using a safety bag (*class II*) [Henriksson et al. 2021; IEC 2010; Klein 1991; Theuretzbacher 1987]. Regarding AI-based heteronomous and autonomous systems, the generic application of offline formal verification seems questionable due to limitations such as the uncertainty and difficulty of explicitly formalizing all safety specifications, rules and constraints required for the safety verification, and the potential high dimensional design space that limits the application of formal verification and brute-force testing approaches [Berghoff et al. 2020; Luckcuck et al. 2019; Torens et al. 2022; Vassev 2016]. A similar limitation applies to online approaches such as the safety bag technique, but in this case, formally specified operational rules can be used to specify safety envelopes (a.k.a., safety monitor, runtime monitor, runtime verification, supervisor, guardian agent, safety layer, safety net) [ASTM 2021; Cofer et al. 2020; Dill et al. [n. d.]; Guiochet et al. 2017; Luckcuck et al. 2019; Rizaldi et al. 2017; Salay et al. 2018; Sarathy et al. 2019; Schirmer et al. [n. d.]; Terrosi et al. [n. d.]]. For example, model checking has already been applied in some specific applications (e.g., AD vehicle overtaking [Rizaldi et al. 2017]) for the development of formally defined *safety envelope* software (*runtime monitor/verification*) [Luckcuck et al. 2019; Rizaldi et al. 2017].

Safety bag and safety envelope type techniques provide a potentially generic safe approach for the adoption of cutting-edge and state-of-the-art AI technology solutions (as a compensatory measure to adapt *Class III* AI technology to *Class II*). However, its use must consider the safety of the system as a whole because, for example, excessive false alarms could lead to new system-level hazards (e.g., cascade errors in systems with multiple safety functions) and should also consider human cognitive limitations (e.g., cognitive overload, oversight and reaction time limitations) [Perez Cerrolaza et al. 2020; Terrosi et al. [n. d.]]. The avionics domain ASTM F3269 [ASTM 2021] standard describes a reference run-time assurance architecture to safely bound the behavior of 'complex functions' integrated in aircraft systems such as UAVs and Unmanned Aircraft Systems (UASs). This architecture implements a safety bag type technique where a *safety monitor* monitors the safe operation of a 'complex function' (e.g., AI-based function) and activates the safe state or switches to a recovery control function [ASTM 2021; Cofer et al. 2020; Dill et al. [n. d.]; Sarathy et al. 2019; Schirmer et al. [n. d.]; Torens et al. 2022] if operating outside established safe operation constraints and rules.

Table 3. Selected *product* safety techniques (*Class I, II*) and example case-studies

Type	UL	*Domain*	*Description*	*Class*	*TAI*	*Technique*
Automatic	*A*	Automotive	Brake pedal state estimation	-	Connectionist	Not specified [Al-Sharman et al. 2021]
		Avionics	Collision avoidance	II	Connectionist	Simulation [Julian et al. 2019, 2016]
		Industrial	Diverse applications	II	Connectionist	Not specified [Lisboa 2001]
		Railway	Interlocking system (SIL4)	II	Optimization	Safety bag [Klein 1991; Theuretzbacher 1987]
	A2, C	Industrial	Sensor diagnostics	-	Connectionist	Diagnostics [Jäger et al. 2018]
Heteronomous and Autonomous	*A*	Automotive	Collision avoidance (ASIL-D)	II	Connectionist	Safety monitor [Al-Khoury 2017]
			Autonomous vehicles platoon Vehicle collision detection	I	Symbolists	Formal verification [Kamali et al. 2017; Törnblom and Nadjm-Tehrani 2018]
			AD vehicle overtaking	I	Not specified	Formal verification [Rizaldi et al. 2017]
		Avionics	Generic safety pattern for complex functions (e.g., navigation and control)	II	Not specified	Safety monitor [ASTM 2021; Sarathy et al. 2019]
			UAVs and UASs	II	Not specified	Safety monitor [Dill et al. [n. d.]; Schirmer et al. [n. d.]]
					Connectionist	Safety monitor [Cofer et al. 2020]
		Industrial	Perception-based solutions for robots	II	Connectionist	Run-time monitor [Rahman et al. 2021]
			Autonomous robots (survey)	I	Not specified	Formal verification [Luckcuck et al. 2019]
		Space	On-board autonomous spacecraft	II	Generic	Safety bag [Blanquart et al. 2004]
	A2, C	Automotive	Vehicle self diagnostics	-	Connectionist	Diagnostics [Yurii and Liudmila 2017]

In addition, Table 4 summarizes the systematic and random errors management techniques described in this Section. At all levels, the overall AI-based safety-critical must comply with the required FuSa, heteronomous, autonomous and AI standards. At the highest *AI system* level, developers define safety assurance cases with the arguments and required evidence needed to justify that the system is safe for its purpose; developers identify and manage uncertainty sources and successfully verify, test and validate the system. *AI item* developers control and mitigate systematic errors using at least the appropriate development lifecycle and techniques, appropriate tools and training platforms, and the obtained ML properties provide sufficient evidence to justify the previous assurance case argumentation. Finally, the underlying platform must avoid, control and mitigate systematic and random errors providing sufficient evidence to the previously defined assurance case argumentation.

Table 4. *Product* - Summary of techniques for systematic and random errors management

Error control and mitigation techniques			**AI-based System** (§4.1)	**AI-item** (§4.2)		
				***Connectionist* NN** ***Connectionist* DL**	**Symbolists**	**Optimization**
Systematic Errors	**AI Development**	*Design Model Training*	Safety assurance case [Abduljabbar et al. 2019; Alexander et al. 2020; Ashmore et al. 2021; Berghoff et al. 2020; Birch et al. 2013, 2020; Bloomfield et al. 2019; Cârlan et al. 2021; ISO 2018; Jenn et al. 2020; Koopman et al. 2019; McDermid and Jia 2020; Picardi et al. 2020; Rudolph et al. 2018; Schwalbe and Schels 2020; Thomas and Vandenberg 2019] VVT [Guiochet et al. 2017; Kalra and Paddock 2016; Kamali et al. 2017; Koopman and Wagner 2016, 2018; Rajabli et al. 2021; Riedmaier et al. 2021; Salay et al. 2018] Uncertainty mgmt. [Alexander et al. 2020; Burton et al. 2020; Koopman and Wagner 2018; McDermid and Jia 2020; Salay and Czarnecki 2019; Schwalbe and Schels 2020; Shafaei et al. 2018; Thomas and Vandenberg 2019; Vassev 2016]	Design and lifecycle [Ashmore et al. 2021; Hains et al. 2018]	Safety bag, adhoc development [Jenn et al. 2020]	Safety bag [Klein 1991; Theuretzbacher 1987], adhoc development
		VVT Model Ver.		Generic [Jacklin et al. 2005; Pullum et al. 2007; Schumann and Liu 2010; Taylor 2006; Taylor et al. 2003b; Zhang et al. 2020] Safety specific [Borg et al. 2018; Huang et al. 2020; Salay et al. 2018; Schwalbe and Schels 2020; Zhang and Li 2020] Formal methods [Huang et al. 2017; Zhu et al. 2021] Metrics [Gharib and Bondavalli 2019; Harel-Canada et al. 2020; O'Brien et al. 2020]		
		Implementation (software, elec.)		FuSa safety standards compliance (see §2.2), e.g., software: IEC 61508-3 7.4.5, 7.4.6		
	ML Properties	*Data Quality*		Dataset properties [Ashmore et al. 2021; Rabe et al. 2021] Engineering requirements [Kuwajima et al. 2020]		
		Auditability		Generic review [Berghoff et al. 2020], Verification [Huang et al. 2017; Kuper et al. 2018]		
		Explainability		Generic surveys [Adadi and Berrada 2018; Barredo Arrieta et al. 2020; Guidotti et al. 2018]	Explicit rules [Jenn et al. 2020; Kamali et al. 2017]	
		Monitorability		Safety bag, Safety envelope [ASTM 2021; Guiochet et al. 2017; Henriksson et al. 2021; Luckcuck et al. 2019; Rizaldi et al. 2017; Salay et al. 2018; Sarathy et al. 2019; Terrosi et al. [n. d.]][ASTM 2021; Cofer et al. 2020; Dill et al. [n. d.]; Sarathy et al. 2019; Schirmer et al. [n. d.]; Torens et al. 2022]	Safety bag	Safety bag [Klein 1991; Theuretzbacher 1987]
		Provability		Formal verification [Corsi et al. 2020; Ehlers 2017; Hains et al. 2018; Jenn et al. 2020; Katz et al. 2017; Pulina and Tacchella 2012; Rudolph et al. 2018; Sun et al. 2019b; Vassev 2016]	Formal ver. [Törnblom and Nadjm-Tehrani 2018]	
		Robustness		Test and adversarial attacks [Akhtar and Mian 2018; Behzadan and Hsu 2019; Kuwajima et al. 2020]		

Error avoidance, control and mitigation techniques		**AI-based System** (§4.1)	**Tools and training platform** (§4.4)	**Execution platform (inference)** (§4.3)		
				Hardware Fwk.	**Software Framework**	**AI Framework**
Syst. & Random Errors	Syst. & Random Errors	Safety assurance case [Cârlan et al. 2021; Jenn et al. 2020; Thomas and Vandenberg 2019]	Generic (not qualified) tools and training platforms	- Generic dev.: Multicore [Mittal and Vetter 2016; Ottavi et al. 2018; Perez Cerrolaza et al. 2020], FPGA [Bernardeschi et al. 2015; Grade et al. 2016] GPU [Perez-Cerrolaza et al. 2022; Santos et al. 2017, 2019] - Specialized dev. [Chen et al. 2019b; Dally et al. 2020; Jouppi et al. 2017; Schuman et al. 2017]: e.g., TPU, NPU, NPU, neuromorphic computing - Custom-designed dev.: e.g., Tesla FSD [Talpes et al. 2020] - Specialized accel.: e.g., DNN [Li et al. 2017]	- Generic AD fwk.: e.g., Apollo [Alcon et al. 2020; Tabani et al. 2019] - Generic: Hypervisor [Burgio et al. 2016; Lampka and Lackorzynski 2019; Perez Cerrolaza et al. 2020]; OS (e.g., Linux [Allende et al. 2021; Bruhn et al. 2020]); Middlewares [Tabani et al. 2020] (e.g., ROS [Luckcuck et al. 2019; Macenski et al. 2022; Tabani et al. 2019], CyberRT [Baidu 2021; Tabani et al. 2019], AUTOSAR [AUTOSAR 2022])	- Adapted / Analyzed / Improved: DL [Biondi et al. 2019; Bosio et al. 2019; Fernandez et al. 2021; Geißler et al. 2021; Li et al. 2017; Ruospo et al. 2020], basic MxM libraries [Fernandez et al. 2021] - Generic Low level libraries: e.g., TensorRT, OpenBLAS, cuBLAS, ATLAS, cuDNN - Safety GPU APIs: OpenGL SC, Vulkan SC [Perez-Cerrolaza et al. 2022]
Safety standard compliance		FuSa, heteronomous, autonomous and AI & safety standards (see §2.2,3.3)				

4.1 AI-based System

Safety assurance cases are commonly used in the development and certification/assessment of traditional FuSa systems to justify that a given safety-critical system is acceptably safe for its purpose, using a structured and evidence-supported safety argumentation [Berghoff et al. 2020; Birch et al. 2013; ISO 2018; Thomas and Vandenberg 2019]. For example, the safety case provides a structured argumentation of systematic and random errors management, from high-level architectural and lifecycle systematic aspects down to the underlying execution platform (see Table 4).

Safety cases are also commonly used for the development and certification/assessment of heteronomous, autonomous, and AI-based safety-critical systems [Berghoff et al. 2020; Bloomfield et al. 2019; Jenn et al. 2020; Koopman et al. 2019; McDermid and Jia 2020; Picardi et al. 2020; Rudolph et al. 2018; Thomas and Vandenberg 2019]. However, for the latter, the safety assurance case should also support the management of *uncertainty-related failures* (see VDE-AR-E 2842-61 [VDE 2021]) inherent to heteronomous, autonomous and (non-trivial) AI-based systems. This AI uncertainty management includes, among others, uncertainty sources identification and uncertainty reduction argumentation [Shafaei et al. 2018; Thomas and Vandenberg 2019]. For example, the safety assurance case arguments of an *AI-item* (§4.2) can be built on claims of high-level properties [Ashmore et al. 2021; Jenn et al. 2020; Rudolph et al. 2018], such as the ML properties defined in Section 2.3 (e.g., *explainability*, *monitorability*, *auditability*, *provavility*), arguments based on specific methods used for uncertainty mitigation during the development phases (e.g., data representativeness of requirements, input space coverage validation) [Alexander et al. 2020; Schwalbe and Schels 2020] and adapt generic argument patterns [Picardi et al. 2020]. However, care must be taken to avoid oversimplifying the safety development challenge to achieving high-level properties with numerical targets and mathematical formulations, without addressing the safety of the system as a whole with associated system hazard elimination [Dobbe 2022; Gharib et al. 2021; Varshney 2016].

The uncertainty management required to reduce *uncertainty-related failures* becomes a key technical aspect to be managed in all AI-related lifecycle phases from the specification to the verification, validation and testing phases. For example, in the specification phase of an AI-based heteronomous/autonomous system, the safety functions (and previous safety goals) can only be specified as 'intended functionality' with a set of high-level goals and objectives [Birch et al. 2020], or iterative partial specifications [Salay and Czarnecki 2019], because it is not generally feasible to fully specify the safety functions (w.r.t. all possible scenarios) with a set of safety requirements, rules, constraints (e.g., [Bergenhem et al. 2015]). This creates a 'semantic gap' [Burton et al. 2020; McDermid and Jia 2020] between the intended functionality and the specified functionality, which sometimes is based on examples where anomalous and edge/corner case examples are a minority. In this context, ensuring that the provided specification provides a correct, accurate and complete representation of the 'intended functionality' is a challenge for the *data management and model training* [Burton et al. 2020]. This challenge can be mitigated by means such as formal verification of safety properties with some degree of uncertainty [Vassev 2016] and safety runtime checkers that during runtime monitor that a set of required constraints are always met (safety operational envelope) [Koopman and Wagner 2018].

On the other hand, the testing and validation of AI-based autonomous systems is still an unsolved key area [Dahm 2010; Helle et al. 2016; Koopman and Wagner 2018; Torens et al. 2022; Weiss 2011], that limits the practical deployment and commercialization of AI-based safety autonomous systems [Dahm 2010; Kalra and Paddock 2016; Koopman and Wagner 2016, 2018] for which current testing techniques designed for 'manned systems' are not directly applicable and sufficient [Thompson 2008], and field testing only based evidences are generally considered not feasible [Kalra and Paddock 2016; Riedmaier et al. 2020]. Therefore, as described by different authors [Kalra and Paddock 2016; Kamali et al. 2017; Koopman and Wagner 2016, 2018] the validation should consider the definition of a strategy with a framework that combines multiple testing techniques and approaches, with the adaptation of existing techniques and the definition of novel techniques specific for AI-based autonomous systems. In fact, the most relevant challenge in heteronomous and autonomous systems test and validation, is the test and validation of the implemented AI solution itself. So, the most common approach for AI-based AD [Kalra and Paddock 2016; Koopman and Wagner 2018; Rajabli et al. 2021; Riedmaier et al. 2020] and collaborative robots [Guiochet et al. 2017] testing and validation relies on simulation frameworks where other AI technology solutions facilitate and automatize the *process* of generating and classifying test scenarios and test cases (see §6.2.3).

Finally, the safety case is not static or defined once, as it requires maintenance updates during the system operational life. And this maintenance update requirement is even more crucial for autonomous systems as they operate in complex and continuously evolving environments [Berghoff et al. 2020; Cârlan et al. 2021].

4.2 AI Item

This section describes safety technical challenges, techniques, and methods associated with the development of AI-based items using different TAIs abstracted from the application-specific requirements and challenges: *connectionist* NN (§4.2.1) and DL (§4.2.2), *symbolists* (§4.2.3), and *optimization* (§4.2.4). For all considered TAIs, AI items are implemented as electronics, software, model configuration and combinations of the previous using traditional FuSa standard technical requirements (e.g., IEC 61508-3 software development guidelines) and deployed on execution platforms (see §4.3).

4.2.1 Connectionist - Neural Network (NN). At the turn of the millennium, there was growing interest in using NNs in safety-critical applications. In particular, the usage of NNs in aerospace applications and compliance with the stringent aerospace safety standards was an active research area. In this section, we report key aspects to consider when NNs trained using supervised learning enter the picture of safety assurance. Note that the content largely applies also to the subsection on Deep Learning (§4.2.2), i.e., NNs for which hidden layers are stacked in attempts to reach human-like performance for perception tasks (e.g., object detection).

Beyond flight controllers, a 2001 review by Lisboa identified a diverse set of industrial use of NNs in safety-related areas [Lisboa 2001]. Examples include power generation and transmission, process industries and transport industries. A common theme among many applications is that NNs were used for automatic control. While Deep Learning (DL) has dominated among connectionists in the last decade, (non-deep) NNs remain a valid and useful approach in many applications. Recent examples of NNs within the scope of this article are diagnostics (e.g., sensor error detection [Jäger et al. 2018], vehicle self diagnostics [Yurii and Liudmila 2017]) and collision avoidance systems in avionics [Julian et al. 2016].

Companies seeking to integrate NNs in safety-critical systems must evolve several practices throughout the development lifecycle [Ashmore et al. 2021; Kurd et al. 2007; Schumann and Liu 2010]. Supervised learning relies on *data* (for *model training* and *model verification*) being treated as first-class citizens during software and systems engineering. As a result, *data management* needs a rigorous process encompassing collection, augmentation, preprocessing, analysis, and maintenance. *Configuration management* needs to expand to cover the data and feature engineering of the iterative work of NN development. And software *architecture specifications* must also encompass fundamental NN design elements and specifics such as activation functions and hyperparameters controlling the learning process. Furthermore, *specifications* and the associated *test specifications* must be augmented to capture the learning behavior of NNs. Lastly, processes must be adapted to align the highly *iterative development of NNs* with the traditional safety engineering of AI-based systems (V-model).

Concerning *model verification*, Taylor *et al.* analyzed early research in progress on the VVT of NNs, with a focus on studies relevant for NASA applications [Taylor et al. 2003b]. There was substantial research funding assigned to the topic in the early 2000s, and the research matured into several books on the topic, e.g., by Taylor [Taylor 2006], Jacklin *et al.* [Jacklin et al. 2005], Pullum *et al.* [Pullum et al. 2007], and Schumann and Liu [Schumann and Liu 2010]. Menzies and Pecheur provided another early VVT survey in 2005 [Menzies and Pecheur 2005].

While the research was conducted around 20 years ago, the main findings remain relevant today. Discussed challenges of NN VVT include state-space explosion, robustness, explainability, co-engineering of NNs and conventional software, and challenges in specifications of ML concepts. Early VVT solution proposals included "formal methods, control theory, probabilistic methods" [Borg et al. 2018], and general process frameworks. Again, several ideas from the early era remain relevant, although some do not scale to the DL approaches that will be discussed in Section 4.2.2.

More recently, Zhang and Li provided a systematic literature review [Zhang and Li 2020] of testing and verification techniques for NN software-based safety-critical control systems. This review complements the earlier work through its selection of 83 publications between 2011 and 2018. However, as this time interval coincides with the breakthrough of DL, which Zhang and Li explicitly include, we highlight that the findings partly fit the next subsection of this paper – the boundary between NN and DL is not sharp. Based on this analysis, the authors identified five high-order themes, i.e., robustness testing, testing toward failure resilience, measuring test completeness, testing for safety assurance, and testing for explainability. Example solution proposals for NN VVT from the last years include: formal methods [Huang et al. 2017; Torens et al. 2022; Zhu et al. 2021] and novel dependability metrics [Gharib and Bondavalli 2019; O'Brien et al. 2020].

4.2.2 *Connectionist - Deep Learning (DL) models.*

The research community acknowledges the potential benefits of using DL in safety-critical applications. In general, developing safety-critical systems that rely on DL shares the same challenges as NNs – as can be seen in Dey and Lee's recently proposed three-layered conceptual framework [Dey and Lee 2021]. However, the fact that contemporary deep NNs can be composed of billions of neurons, organized into complex architectures, further amplifies all challenges. Several VTT practices mandated by FuSa become less effective, e.g., code reviews matter less if the logic resides in the training data [Salay et al. 2018] and the value of adequacy testing metrics is questionable [Harel-Canada et al. 2020].

Still, the representation learning offered by DL has enabled several breakthroughs during the 2010s and trained DL models have outperformed human performance in a range of restricted tasks. From the perspective of this review, the use of DL has disrupted computer vision and enabled perception systems able to generalize to diverse operational contexts. Advances in the automotive industry have been particularly prominent, with DL being a key enabler for AD, and in various ADAS such as automatic emergency braking and lane keeping assistance [Beglerovic et al. 2018; Chen et al. 2021; Kiran et al. 2021]. Examples of DL use in the aerospace sector include collision avoidance systems [Julian et al. 2019].

Engineering a trustworthy DL-based system is largely about managing a dynamic *ML workflow* with iterative updates. First, the development of a DL system is an experimental and highly iterative process where the "Changing Anything Changes Everything" principle reigns [Sculley et al. 2015], i.e., all data science activities are intertwined and implications of minor changes are hard to foresee. Second, DL-based systems are typically deployed in dynamic operational environments in which conventional software systems would be insufficient. Third, the AI systems themselves can be dynamic post-release if retraining of internal models is enabled (see §5). Thus, integrating automated quality assurance throughout the product lifecycle is essential. Key automation steps, sometimes explained in the context of ML operations (MLOps) tools [Borg 2022; Granlund et al. 2021], include data version control and experiment tracking to support the iterative DL development and solutions for runtime monitoring [Rahman et al. 2021], e.g., to support detection and management of data drifts.

Model verification explicitly targeting DL-based systems is currently a highly active research topic. Borg *et al.* provides an automotive domain-specific review of verification and validation of DL-based solutions [Borg et al. 2018]. A similar study was reported by Schwalbe and Schels [Schwalbe and Schels 2020]. Zhang *et al.* found that most academic studies focused on testing the correctness and robustness, while qualities such as interpretability, efficiency, and privacy are much less studied [Zhang et al. 2020]. Riccio et al. concluded in their systematic analysis that test input and test oracle automated generation for DL systems was the most active research topic for DL *model verification* [Riccio et al. 2020]. Huang et al. provided a DNN specific survey [Huang et al. 2020] covering verification, testing, adversarial attack and defense, and interpretability aspects.

Regarding ML properties for the construction of safety assurance cases, there is a rich variety of research contributions applicable to both NNs and DL models:

- *Data Quality*: The training data implicitly specify the intended functionality, rules and constraints. So data quality is of paramount importance as described by Ashmore et. al [Ashmore et al. 2021], and the *data management* phase must produce datasets that exhibit at least properties such as: relevance, completeness, balance and accuracy [Ashmore et al. 2021; Rabe et al. 2021]. Training data is split for *model training* and *model verification*. In generic applications, the split (e.g., 80%-20%) can be performed randomly, but for safety-critical systems the split shall consider aspects such as: the training data shall completely specify the intended functionality, sufficient representation of edge/corner cases in both training and test data, and the deviation between training/test and operational data shall be minimized [Kuwajima et al. 2020].
- *Explainability*: Several surveys and reviews summarize the high research activity that addresses the NN and DL models explainability challenge [Adadi and Berrada 2018; Barredo Arrieta et al. 2020; Guidotti et al. 2018; Torens et al. 2022]. One can argue that a model is explainable if it is interpretable, and Rudin [Rudin 2019] elaborates on why an interpretable model lowers complexity and thus are to be preferred compared to a model that can not explain the behaviour of a NN or DL solution.
- *Provability*: Multiple research contributions address *provability* of NNs and DL models by means of formal verification [Corsi et al. 2020; Ehlers 2017; Hains et al. 2018; Jenn et al. 2020; Katz et al. 2017; Pulina and Tacchella 2012; Rudolph et al. 2018; Sun et al. 2019b; Vassev 2016]. However, formal verification is (nowadays) limited to moderate size NNs and certain architectures [Jenn et al. 2020; Katz et al. 2017; Torens et al. 2022]. For example, the Reluplex method has been used to formally verify ReLu (Rectified Linear Unit) activation properties of a NN with 300 nodes [Jenn et al. 2020; Katz et al. 2017].
- *Robustness*: Robustness and resiliency can not be evaluated in the *model verification* with (only) test data [Kuwajima et al. 2020]. Nonetheless, this is an active research area [Behzadan and Hsu 2019; Kuwajima et al. 2020] under the topic of adversarial attacks (security) [Akhtar and Mian 2018]. The final objective is to analyze and develop solutions that are robust/resilient with respect to (adversarial) perturbations.
- *Auditability*: Huang et al. propose a framework for the automated safety verification of DNNs made classification decisions [Huang et al. 2017]. Verification is also put forward by Kuper et al. [Kuper et al. 2018] as a viable solution to confirming that NNs behave as intended. In addition, they further suggest to create and use design principles for NNs that produce DNNs that are more amenable to verification [Kuper et al. 2018]. The European Union (EU) has proposed an AI act [EU 2021] that aims to propose a set of harmonized rules on AI. Hence, the work by Kuper et al. [Kuper et al. 2018], as well as contributions by other scholars presented in this survey, may become building blocks to conform with the proposed AI act.

4.2.3 Symbolists. Decision trees can provide explanations and understandability of decisions made by black-box type AI-based items [Guidotti et al. 2018] so that the user is aware of the rationale for decisions and takes control of the safety system if necessary [Jenn et al. 2020] (*A2*). For example, decision trees can provide runtime explanations of decisions made by an ML-based co-pilot to an aircraft pilot, who must understand them and react safely in case of wrong decisions [Jenn et al. 2020]. An equivalent approach can be used offline, during the *product* development.

Random forests can also learn safe operation rules from training data to implement safety functions such as vehicle collision detection (*A2*) [Törnblom and Nadjm-Tehrani 2018]. Furthermore, whenever feasible, safe operation rules can also be explicitly expressed using formal symbolist languages in *rationale agents* (*A1*) for diverse applications such as autonomous vehicle platooning [Kamali et al. 2017]. Both approaches provide support for *explainability* (white-box), *auditability* and *provability* (formal verification) requirements. Finally, Törnblon et al. analyze and propose a method and tool for the formal verification of random forests [Törnblom and Nadjm-Tehrani 2018].

4.2.4 Optimization. FuSa-compliant optimization algorithm-based safety-critical systems can be developed with the safety bag compensation measure [IEC 2010] (*Class II*). The optimization function executes a safety related function that is not subject to a complete safety certification process and development, because a run-time safety bag is developed and certified, which ensures that provided results are safe for its purpose and performs associated safety actions if not (e.g., safe state activation). This approach can be used whenever the optimization function cannot be formally verified at design time, or whenever the safety development of optimization software and tools in compliance with FuSa standards requirements is considered not feasible. For example, this safety technique was already used in the 80s to develop a SIL4 railway signalling system that provides optimized and safe results [Klein 1991; Theuretzbacher 1987].

4.3 Execution Platform (Inference)

The implementation of AI items as embedded software/electronic components with associated model configurations must follow traditional FuSa standard requirements (e.g., software: IEC 61508-3 7.4.5, 7.4.6). Nonetheless, a common approach is to make use of existing execution platforms rather than developing complete ad-hoc implementations. Execution platforms are commonly composed of a hardware platform with High Performance Computing (HPC) capability (e.g., Graphics Processing Unit (GPU)), a software framework (e.g., hypervisor, AUTOSAR, Robot Operating System (ROS)) and an AI software framework (e.g., YOLO, Tensor Flow). And this *execution platform* is the safety computing channel, or one of the safety computing channels of the safety-critical system architecture (e.g., [Yoshida 2020]), developed in compliance with applicable FuSa standard requirements. Additionally, in some specific applications, such as AD [Talpes et al. 2020] and UAV systems (e.g., drone) [Dill et al. [n. d.]; Liu et al. 2020], execution platforms must meet Size, Weight, and Power (SWaP) constraints while providing the required computing performance and FuSa compliance support [Perez-Cerrolaza et al. 2022; Perez Cerrolaza et al. 2020].

As summarized in the survey by Perez-Cerrolaza et al. [Perez-Cerrolaza et al. 2022], the mitigation of random errors by means of evaluation and deployment of diagnostics and fault tolerance mechanisms, is an active research field for DL software frameworks and high-performance computing devices such as GPUs [Santos et al. 2017, 2019], FPGAs [Bernardeschi et al. 2015; Grade et al. 2016], multi-core devices [Mittal and Vetter 2016; Ottavi et al. 2018; Perez Cerrolaza et al. 2020] and specialized accelerators (e.g., DNN [Li et al. 2017]). Or even the definition of specialized software architectures for the development of DL technology-based safety-critical systems [Biondi et al. 2019] and built-in integration of diagnostics measures in software frameworks [Fernandez et al. 2021]. The analysis and error mitigation in the DL algorithms and software implementation is also an active research field [Bosio et al. 2019; Geißler et al. 2021; Li et al. 2017; Ruospo et al. 2020]. Unlike non-DL software, for which fully deterministic and accurate results are expected, DL items often deliver approximate and stochastic results. Hence, error detection is a key challenge for DL items due to multiple challenges: (i) determining whether a result is fault-free is convoluted for a stochastic item that may use also some random numbers as input and whose intrinsic error rate is non-negligible (e.g., object misclassification rates); and (ii) if the DL item inherits a high-integrity level that cannot be diminished with item decomposition (e.g., using a non-DL item that inherits safety requirements and relieves the DL item), then diverse redundancy may lead to different fault-free results owing to the source of diversity (e.g., different random numbers, different training data, different order of computation causing different rounding of results).

4.3.1 Hardware Platform. As the computing power required to execute AI algorithms such as DL models continues to increase, their deployment is commonly based on generic HPC devices (e.g., GPUs, FPGAs, multi-core devices) [Ma et al. 2020], specialized accelerators (e.g., Tensor Processing Units (TPUs), Network Processing Units (NPUs), neuromorphic computing) [Chen et al. 2019b; Dally et al. 2020; Jouppi et al. 2017; Schuman et al. 2017] and custom-designed devices (e.g., Tesla FSD [Talpes et al. 2020]) often including specialized accelerators (e.g., DNN accelerator). With respect to FuSa-compliance, the deployment of safety AI items in generic HPC devices is a feasible approach that needs to take into consideration several technical challenges (e.g., random errors, systematic errors, common cause failures) required by associated FuSa standards (e.g., ISO 26262-11, IEC 61508-3 Annex F), as summarized in the specialized surveys for multi-core devices [Perez Cerrolaza et al. 2020], GPUs [Perez-Cerrolaza et al. 2022] and FPGAs [Bernardeschi et al. 2015].

4.3.2 Software Framework. Available research and open-source AD specific software frameworks (e.g., Apollo [Alcon et al. 2020]), have some limitations with respect to FuSa compliance that limit their applicability, owing to their use of middlewares and operating systems easing decoupling by means of interfaces to subscribe services to events at the expense of an abuse of pointers, unobvious control flow, and deep if-conditional nesting [Tabani et al. 2019].

These specialized autonomous AD software frameworks, along with traditional FuSa and autonomous safety-critical systems, can be built using generic software frameworks such as domain-specific middlewares, hypervisors and Operating Systems (OSs) [Burgio et al. 2016; Martinez et al. 2018; Perez-Cerrolaza et al. 2022; Perez Cerrolaza et al. 2020]. For example:

- Middlewares and domain-specific standard frameworks, including ROS [Luckcuck et al. 2019; Macenski et al. 2022], Apollo's CyberRT [Baidu 2021], and AUTOSAR [AUTOSAR 2022], enable the development of AD frameworks and the use of HPC platforms. On the one hand, some frameworks such as ROS and CyberRT, used along with different versions of Apollo, ease the implementation of AD frameworks, but are not yet integrated with appropriate hypervisors, use interfaces challenging certification (e.g., abundant use of pointers, including function pointers) [Tabani et al. 2019], and do not provide native time predictability [Alcon et al. 2020]. On the other hand, platforms such as AUTOSAR Adaptive are intended to enable the deployment of automotive systems on HPC platforms, but, to our knowledge, they have not been used yet as part of AD frameworks.
- Virtualization technology (e.g., hypervisors) supported by modern multi-core and GPU devices enable the safety compliant integration of software partitions with even different safety criticality levels [Burgio et al. 2016; Perez Cerrolaza et al. 2020]. However, to our knowledge, AD frameworks do not yet build on hypervisors, partly because those frameworks require HPC devices that may miss the support needed by hypervisors to effectively implement partitioning. Hypervisor technology is, however, planned to be used in some forthcoming hardware platforms and use cases [Lampka and Lackorzynski 2019].
- There is an increasing interest in Linux for critical systems (e.g., Automotive Grade Linux) and multiple research and industrial project initiatives aim to enable Linux for the development of safety-critical software [Allende et al. 2021]. For example, Linux is assessed for space systems including HPC SoCs equipped with ML accelerators [Bruhn et al. 2020].

4.3.3 AI Framework. A number of AI frameworks, Keras, Pytorch, TensorFlow, MXNet, Theano and Caffe, are highly popular for generic AI applications. Often, DL models are mapped onto those generic frameworks, which are often selected based on characteristics such as user friendliness (often related to the existence of a high-level API), modularity, efficiency, and the like [Tabani et al. 2020].

AI frameworks may already use primitives for mathematical operations used for DL models, such as Generalized Matrix-Matrix multiplication (MxM), among others. Those primitives are then instantiated for the specific target platform using platform-specific and/or low-level libraries such as TensorRT, OpenBLAS, cuBLAS, ATLAS, and cuDNN, to name a few.

Whether AI frameworks implementation complies with domain-specific standards relates to the specific implementation of the primitives used. Generally, those implementations do not provide any specific safety support, but some APIs and other works provide alternative implementations with safety requirements in mind for CPUs [Fernandez et al. 2021] (e.g., embedded diagnostics) and GPUs [Perez-Cerrolaza et al. 2022] (e.g., with specific APIs such as OpenGL SC and Vulkan SC).

4.4 Tools and Training Platform

There is a rich and dynamic variety of generic frameworks (e.g., TensorFlow), infrastructure (e.g., GPU servers, cloud infrastructure) and tools for the development of generic AI solutions (e.g., *model training*) [Borg 2022; Granlund et al. 2021; Nguyen et al. 2019]. Nonetheless, these generic solutions were not designed with safety standards compliance requirements such as tool qualification. So, this is a potential source of systematic errors (e.g., tool and process errors) and hardware random errors (e.g., training data corruption, GPU random error during *model training*) not generally addressed in research contributions [Granlund et al. 2021; TUVR 2022].

5 RUNTIME - AI ONLINE LEARNING/ADAPTATION

This section describes selected techniques and methods for the AI online learning/adaptation of AI-based safety-critical systems (*runtime*). By default, *runtime* adaptation leads to a 'one of a kind' safety-critical system instantiation that, if unconstrained, is beyond the scope of current and novel safety standards [Jacklin et al. 2005; Koopman and Wagner 2017]. For example, in this scenario, an AD system might adapt and learn new behaviors [Ronald 2013] that were not considered, verified and validated in the offline development and safety certification/assessment process [Koopman and Wagner 2017]. And this adaptation could even be implemented as continuous [Alexander et al. 2020] and lifelong learning [Parisi et al. 2019]. Thus, the 'one of a kind' safety-critical system instantiation may differ from the originally certified/assessed system.

However, as summarized in Table 5, it is feasible to consider constrained AI runtime learning/adaptation approaches (§5.1), for which correctness and completeness of all possible variants is considered in the safety-critical system development process and safety certification/assessment.

Table 5. Selected *runtime* safety techniques and example case-studies

Type	**UL**	***Domain***	***Description***	***Class***	***TAI***	***Technique***
Automatic, Heteronomous or Autonomous	*C*	Avionics	Intelligent Flight Control System	II	Connectionist	Safety bag [Osborne et al. 2021; Taylor et al. 2003b]
	A	Avionics	Gas turbine aero engine control Generic adaptative control system	I I, II	Connectionist, Symbolist Generic, Connectionist	Safe adaptation [Kurd and Kelly 2004, 2005] Safe adaptation [Jacklin et al. 2005]
	A	Aerospace	Adaptative guidance	I, II	Connectionist	Limited adaptation [Johnson et al. 2001]
	A, C	Industrial	ILC-based hydraulic machinery	I, II	Optimization	Limited actuation [Trojaola et al. 2020]

5.1 Runtime Learning/Adaptation

Table 5 summarizes the most relevant techniques and methods selected from research contributions that focus on AI runtime learning/adaptation approaches for developing dependable or safety-critical systems: safety bag (§5.1.1), safe adaptation (§5.1.2), limited adaptation (§5.1.3), limited force (§5.1.4) and 'library based offline' (§5.1.5). Some selected research contributions describe techniques for developing dependable systems and not explicitly safety-critical systems. However, these techniques are adaptable to safety standard requirements; thus, this section describes and adapts them. Finally, it is assumed that the implementation of described techniques meets basic safety assumptions [Alexander et al. 2020]: e.g., it is an authorized adaptation, the adaptation has a well-defined process (e.g., trigger command, update time) and implements basic error detection/control measures.

5.1.1 Safety bag. The previously explained *Class II* safety bag technique (a.k.a., safety monitor), can also ensure that the outputs provided by the AI-item subject to runtime learning/adaptation are safe. As previously explained, the safety bag becomes the safety function and the AI-item becomes a non-safety function (*C*). For example, the avionics Intelligent Flight Control System (IFCS) aims to safely optimize aircraft flight performance with two NNs, one trained offline and the second one while the aircraft is in operation (Online Learning Neural Network (OLNN)) [Taylor et al. 2003a]. And the system runs two safety monitors, one for each NN, where the OLNN safety monitor checks the safeness of the provided outputs. Another example is AI-generated online trajectory monitor of (slow-dynamic) autonomous systems using techniques such as Nonlinear Model Predictive Control (NMPC) [Osborne et al. 2021].

5.1.2 Safe Adaptation. The *safe adaptation* technique requires both the AI-item and the runtime learning/adaptation algorithm to be safety-compliant. This is because both must perform safety functions, safe inference and safe runtime learning/adaptation. For example, Kurd et al. [Kurd and Kelly 2004, 2005] describe a safety-critical 'gas turbine aero engine control' based on a hybrid TAI (*connectionist*, *fuzzy*) that performs runtime adaptation to provide safe control while safely adapting to the engine degradation and environmental change. Additionally, Jacklin et al. [Jacklin et al. 2005] describe challenges and example techniques for the development of safe adaptive control solutions using learning algorithms such as NNs (e.g., learning convergence, speed of learning convergence, learning algorithm stability).

5.1.3 Limited Adaptation. The *limited adaptation* technique safely constraints the internal runtime learning/adaptation, either through a safety compliant adaptation (*Class I*) or a safety bag that checks the adaptation outcome (*Class II*, see §5.1.1). For example, Johnson et al. [Johnson et al. 2001] describe using NNs to perform adaptive control of an autonomous launch vehicle guidance system. The system uses an adaptive NN-based error cancellation algorithm to cancel the control error due to differences between the actual vehicle dynamics and the design-time vehicle model, with a "bounded weight update law" that safely constrains the runtime learning/adaptation.

5.1.4 Limited Actuation. The *limited actuation* technique ensures that the AI-item subject to runtime learning/adaptation cannot exceed given dangerous output actuation values (e.g., excessive force, energy, voltage). This could be implemented in different ways, such as design-time constraints (e.g., limited input energy leads by design to limited output energy), AI-based safety function that guarantees a limited actuation (*A1*, *Class I*) or a safety bag that monitors and ensures that output actuation values are within safe limits (*C*, *Class II*, see §5.1.1).

In particular, the Iterative Learning Control (ILC) approach is used in dependable industrial control systems such as robots and machinery. ILC [Bristow et al. 2006] aims to optimize the execution of repetitive tasks by learning from previous executions. For example, Trojaola *et al.* [Trojaola et al. 2020] propose an ILC algorithm for hydraulic machinery systems that can be used online to adapt and learn the compensating force required to reduce overshoot and settling time even with unknown knowledge of the valve dynamics. In this scenario, a runtime monitor can be used to monitor and ensure that the learning/adaptation actuation results are safely limited (e.g., compensatory force, dynamic behavior, settling time [Trojaola et al. 2020]).

5.1.5 Library-Based Offline. The *library-based offline* technique defined for nonlinear control systems [Osborne et al. 2021] can be translated in the safety-critical domain as a library of possible configurations defined and assessed offline, to which the system can transition during runtime (*Class I*). This is the adaptation of a common approach used in the development of traditional safety-critical systems, where all possible configuration and operational modes are defined and assessed offline (e.g., normal and degraded modes of operation).

6 PROCESS - AI-BASED DEVELOPMENT ASSISTANCE

This Section describes AI-based offline techniques and methods that support and facilitate the *traditional safety engineering* of safety-critical systems (§6.1) and the *AI safety engineering* of *AI items* (§6.2). For the latter, developers use AI-based solution(s) to develop *AI item(s)* (e.g., DL-based perception item tested using test scenarios defined with Bayesian optimization).

There is a considerable amount of research contributions proposing AI-based techniques to support and assist non-safety-related software developers [Feldt et al. 2018; Martínez-Fernández et al. 2022] (e.g., software test automation [Hourani et al. 2019; Lima et al. 2020; Ramanathan et al. 2016; Salvado 2019]). However, these generic contributions (*Class III*) cannot be directly used to develop safety-critical systems because they do not comply with the strict method, process and tool qualification requirements imposed by safety standards. Nevertheless, these contributions could complement traditional methods and techniques that already meet the requirements of safety standards. But, the intended use of these contributions would not yet be safety-related and are considered outside the scope of this survey.

On the other hand, as summarized in Table 6, there are multiple research contributions proposing AI-based solutions to support and assist developers of safety-critical systems. It is worth noting that the amount of research contributions focusing on *AI safety engineering* is higher than those focusing on *traditional safety engineering*, due to the novelty of the challenge posed by the former and the diversity and rich variety of 'problems' (challenges) to solve (e.g., model verification). Furthermore, this diversity and rich variety of challenges require the use of a diverse and rich variety of AI-based solutions that cover all TAIs summarized in Section 3.1.

Finally, we should also mention that AI solutions are also commonly integrated into hardware ASIC design tools, FPGA synthesis tools and software compilers [Huang et al. 2021; Leather and Cummins 2020; Wu and Xie 2022]. And manufacturers for safety-critical systems already address systematic errors through mass-produced electronic integrated circuits requirements (e.g., IEC 61508-2 §7.4.6.1) and tool qualification requirements (e.g., IEC 61508-4 §3.2.11, ISO 26262-8 §11.4.5/6).

6.1 Traditional Safety Engineering

Research contributions that focus explicitly on *traditional safety engineering* of safety-critical systems are fragmented and scarce (see Table 6a). This Section describes some selected example research contributions following the V-model structure (see Figure 2a).

6.1.1 Specification, Design and Implementation. A single systematic error in the requirements, design or implementation phase could directly lead to a fatal consequence. So, safety standard requirements (e.g., tool qualification) are stricter and research contributions that explicitly target safety-related systems are fragmented and scarce (both *class I* and *class II*). For example:

- Specification: Natural Language Processing (NLP) solutions can be used for safety assessment and analysis of textual requirements (e.g., hazard identification [Daramola et al. 2013]) with human safety expert verification of the proposed results as a compensation measure to become *Class II*.
- Design and implementation: Optimization algorithms and formal verification techniques can be combined to facilitate the design of FuSa-compliant safety functions [Perez et al. 2021]. The optimization algorithm proposes an optimized design for a given criterion, and the formal verification verifies compliance with all applicable safety rules and constraints. To do this, the safety requirements that define the safety rules and constraints are expressed both formally for the formal verification and semi-formally for the optimization

Table 6. Summary of AI-based development assistance solutions for safety-critical systems (*class I and II*)

Lifecycle (Phase)	**Usage Purpose**	**Type of AI (TAI)**
Spec.	Hazard identification	Connectionist (NLP), symbolic (ontologies) and analogizer (CBR) [Daramola et al. 2013]
	SIL evaluation	Symbolic (Fuzzy [Ouazraoui and Nait-Said 2019; Sallak et al. 2006]) Bayesian (DBN [Simon et al. 2019])
Design	Design optimization	Optimization (ACO, EDA, ILS [Gheraibia et al. 2018; Perez et al. 2021])
Test Validation	Test definition automation for MC/DC coverage	Connectionist (NN [Čegiň and Rástočný 2020]) Optimization (GA [El-Serafy et al. 2015]) Symbolic [Godboley et al. 2021]

(a) *Traditional safety engineering* (V-model)

Lifecycle (Phase)	**Usage Purpose**	**Type of AI (TAI)**
Data Mgmt.	See *model verification*: test definition automation	
Model training	Design optimization (AutoML)	Reinforcement learning [Hutter et al. 2019; Waymo 2019] Bayesian [Hutter et al. 2019]
Model Verification	Test definition automation	Connectionist (RNN [Jenkins et al. 2018], GAN [Krajewski et al. 2018], autoencoder [Krajewski et al. 2018]) Bayesian [Abeysirigoonawardena et al. 2019; Akella et al. 2020; Gangopadhyay et al. 2019; Jesenski et al. 2019; Wang and Zhao 2018] Optimization [Albaba and Yildiz 2022; Ben Abdessalem et al. 2018; Du and Driggs-Campbell 2019; Mullins et al. 2018; Tuncali et al. 2020] Symbolists [Bagschik et al. 2018; Li et al. 2020]
	Test classification automation	Connectionist (CNN [Beglerovic et al. 2018], RNN [Beglerovic et al. 2018]) Symbolic (random forest [Kruber et al. 2019])
	Fault injection	Bayesian [Jha et al. 2019]
	Rule extraction	Symbolist (fuzzy [Kurd and Kelly 2004], tree [Jacobsson 2005, 2006])
	Quantify uncertainty	Bayesian [Fan et al. 2020; Gal 2016; Kendall et al. 2015; Kendall and Cipolla 2016]

(b) *AI safety engineering* (for ML)

process. And, as the result is formally verified (*Class I*), state-of-the-art non-safety related AI software tools, engineers and methods can be used for the design optimization proposal activity.

6.1.2 Verification, Validation and Testing (VVT). Software test automation is an active research area for non-safety related systems [Hourani et al. 2019; Lima et al. 2020; Ramanathan et al. 2016; Salvado 2019]. Concerning safety-critical systems, the most relevant challenge addressed is the generation of automated test data and test cases to achieve the level of safety software test coverage requested by safety standards [IEC 2010], such as the Modified Condition/Decision Coverage (MC/DC) percentage levels. AI algorithms can facilitate achieving the recommended 100% MC/DC criteria for software unit test activity (IEC 61508 Table B.2), reducing the safety engineering effort required to perform a detailed analysis of all software code paths and test data combinations that could lead to testing all software code statements and execution branches. To that end, symbolic [Godboley et al. 2021], NN [Čegiň and Rástočný 2020] and GA [El-Serafy et al. 2015] solutions have been proposed for test data generation and the achieved MC/DC value can be potentially verified with Commercial Off-The-Shelf (COTS) qualified tools [Godboley et al. 2021] (*class I*).

6.2 AI Safety Engineering

Concerning the *AI safety engineering* of AI-based (sub)systems and items, most research contributions describe ML-based solutions for connectionist-based *products*. So this Section follows the ML workflow described in Section 3.5 and Figure 2b. As summarized in Table 6b, research contributions that target the data management and model learning phases are scarce, and solutions that target the model verification phase are more abundant specially for the VVT activities of heteronomous/autonomous systems.

6.2.1 Data Management. As stated in the generic survey of software engineering for the development of AI-based systems, “data-related issues are the most recurrent type of challenge” with limited mitigation techniques described in the surveyed papers [Martínez-Fernández et al. 2022]. This generic statement can also be extended to the safety-critical and AI-based *process* niche, which primarily focus on the automated generation of test data and scenarios as described for *model verification* (see §6.2.3).

6.2.2 Model Learning. Automated ML (AutoML) refers to the methods, techniques, and processes that aim to automate the development of ML models [Ashmore et al. 2021; Berghoff et al. 2020; Hutter et al. 2019]. For example, selecting optimal DL hyperparameters for developing ML models for autonomous driving tasks is time-consuming for engineers. And autoML has been (functionally) evaluated as a successful approach for the design automation of perception tasks ML models, with results that outperformed the ones obtained by trial-error approaches by experienced engineers (higher *accuracy*, less latency) [Waymo 2019]. For that purpose, the autoML can use a variety of technical approaches such as random search, reinforcement learning approaches and Bayesian optimization to explore the design space [Hutter et al. 2019; Waymo 2019]. However, AutoML-based design space exploration requires higher computational resources than human-guided designs and training infrastructure scalability becomes a technical concern [Hutter et al. 2019]. Also, there is still a lack of both safety standard requirements to guide the autoML systematic error reduction and research contributions proposing methods or techniques in this line.

6.2.3 Model Verification. AI technology also plays a crucial role in the scalability, efficiency and automation of AI-based items/systems' testing and validation *processes* (see §4.1). The (pseudo) manual definition of test scenarios and test cases is considered not feasible or scalable for heteronomous/autonomous systems [Gangopadhyay et al. 2019; Wang and Zhao 2018]. Ma et al. [Ma et al. 2022] provide an up-to-date review of AI in the VVT of AD systems, dividing the works into scenario-based testing, formal verification and fault injection testing. This is an active area of research [Rabe et al. 2021; Riccio et al. 2020], with a rich variety of TAIs that can be used for the automation of these VVT tasks, and associated *model verification* activities:

- *Connectionist* solutions: DL technologies can "discover intricate structures well in high-dimensional data and learn the idea of correct representation of data" [Al-Sharman et al. 2021; Sun et al. 2019a]. Therefore, they are commonly used for the unsupervised modeling and generation of test scenarios/cases, such as vehicle maneuver modeling using autoencoder and Generative Adversarial Network (GAN) solutions [Krajewski et al. 2018]. One advantage of this approach is that in both cases, the learned model has been trained to generate trajectories that even the discriminator (for GAN) is not able to distinguish between real life or synthetic trajectories [Krajewski et al. 2018]. Another common approach is the generation of test scenarios using RNNs (e.g., accident scenarios [Jenkins et al. 2018]) or scenario classification using RNNs and CNNs [Beglerovic et al. 2018]. Furthermore, the number of scenarios explored can be increased dramatically through the use of deep Q-learning [Albaba and Yildiz 2022].
- *Bayesian* solutions [Abeysirigoonawardena et al. 2019; Gangopadhyay et al. 2019; Jesenski et al. 2019; Zhao et al. 2018] are also commonly used for the unsupervised generation of test data, test cases and test scenarios using the learned probability distribution for the given problem to generate variants. For example, generation of intersection scenes [Jesenski et al. 2019] and traffic scenarios [Wang and Zhao 2018]. And for a given test scenario, Bayesian optimization can be used to learn from observed system outputs and define test cases that could violate predefined safe operation boundaries [Gangopadhyay et al. 2019]. Furthermore, Bayesian techniques can also be used for classification (e.g., a nonparametric Bayesian approach has been used to cluster adversarial policies [Chen et al. 2022]). Finally, Bayesian solutions have also been proposed for fault injection (e.g., a Bayesian fault injection framework uses "causal and counterfactual reasoning about the behavior under a fault" to find faults/errors) [Jha et al. 2019].
- *Symbolic* solutions: Ontology-based combination "is an essential approach to generate testing scenarios, which combines scenario entities based on ontology theory for the primary goal of coverage" [Bagschik

et al. 2018; Li et al. 2020]. And random forests are commonly used for unsupervised test scenario clustering and classification [Kruber et al. 2019].

- *Optimization* solutions: Search techniques have been widely applied for testing [Saeed et al. 2016], for example multiobjective search [Ben Abdessalem et al. 2018], Monte Carlo Tree Search (MCTS) [Du and Driggs-Campbell 2019], adaptive search [Mullins et al. 2018], and requirements-driven test generation automation with simulated annealing [Tuncali et al. 2020]. Finally, Fan et al. [Fan et al. 2020] and Fisac et al. [Fisac et al. 2019] describe Bayesian model learning solutions via Bayesian NNs or statistical Gaussian processes, which support the optimization and safe control design of adaptable safety-critical systems with control stability and safe limits.

As the available offline computing power continues to increase, the use of statistical testing approaches supported by automated test scenarios/cases generation that obtain sufficient statistical representativeness could be a new approach to explore for AI-systems [Jenn et al. 2020], in analogy to probabilistic WCET [Cazorla et al. 2019] and probabilistic testing approaches for Linux-based safety systems [Allende et al. 2021]. AI technology (*process*) can also be used for the verification of AI-items. For example, *symbolist* trees can be used for rule extraction of RNN-based items for both understandability and verification purposes [Jacobsson 2005], and *Bayesian* methods are proposed for the uncertainty quantification of DL-based safety applications [Fan et al. 2020; Gal 2016; Kendall et al. 2015; Kendall and Cipolla 2016].

7 TRUSTWORTHINESS

As stated in the standard VDE-AR-E2842-61 [Putzer et al. 2021; VDE 2021], *trustworthiness* "has not generally accepted definition" at least in the context of AI-based safety-critical systems. Nonetheless, if we analyze in detail the standard VDE-AR-E2842-61 [Putzer et al. 2021; VDE 2021], technical reviews in the field of safety and AI [Burton et al. 2020; Dobbe 2022; Huang et al. 2020] and generic AI guidelines (e.g., "Ethics guidelines for trustworthy AI" [EU 2019]), we can identify at least three dimensions applicable to AI-based safety-critical systems: engineering (§7.1), ethics (§7.2) and legal dimensions (§7.3). Thus there is a multidisciplinary collaboration requirement to address all trustworthiness dimensions (e.g., engineering, philosophy, ethics, social sciences, law), along with a multi-agent collaboration requirement among all relevant actors such as companies, governments, legislators, regulators, standardization organizations, certification bodies, academia and society in general.

Indeed, the increasing importance of trustworthiness in the development of AI-based safety-critical systems is emphasized in the VDE-AR-E2842-61 standard with the Trustworthiness Performance Level (TPL) (TPL 0-4) definition that requires trustworthiness attributes traceability through the AI-based system development activities, design patterns supporting the verification of AI properties, and compliance with specific techniques/measures pending definition details in the current draft [VDE 2021].

7.1 Engineering Dimension

The engineering dimension must cover at least non-functional properties such as robustness, dependability (reliability, availability, maintainability, safety) [Avižienis et al. 2004], and cybersecurity [TUVR 2022; VDE 2021]. Previous sections (§4, §5, §6) have already addressed the safety engineering dimension of AI-based safety-critical systems. And implicitly, to some extent, robustness and dependability aspects relevant to the scope of the given survey. Also note that, the engineering trustworthiness relies on previously described safety assurance cases (see §4.1) that provide a structured safety engineering argumentation with associated evidences and risk assessment [Bloomfield et al. 2019].

Concerning cybersecurity, the life cycle of AI is complex by nature, and it involves several phases such as planning, data management, model training, model evaluation and operation. This represents a vast attack surface that can take place in each phase, posing a threat to both security and safety ("no safety without security"). In the planning stage, developers are a candidate to suffer social attacks that can negatively influence the whole process. The data management and model training processes are the pillars for building models, and poison attacks [Eicher et al. 2020; Quiring et al. 2020] can impact models in different and relevant aspects, such as accuracy in operation [Biggio and Roli 2018]. In order to address these threats it is necessary to plan a defense strategy at two levels: data and people. Concerning information, the defense aims to prevent information stealing and adversarial attacks [Biggio et al. 2012] using strategies such as differential privacy [Du et al. 2020], data encryption, adversarial training, standardization and verification of data quality, supply chain, and training process [Madry et al. 2018; Metzen et al. 2017; Yang et al. 2020], among others [Chen et al. 2019a]. On the other hand, regarding people, awareness and training programs for detecting social manipulations are recommended. Finally, in evaluation and operation, several attacks can take place in different aspects, such as hardware level attacks [Tran et al. 2018], adversarial attacks [Sharif et al. 2016], inference attacks [Shokri et al. 2017] and stealing of models [Tramèr et al. 2016]. In order to address these threats, system developers can use different prevention techniques, such as feature squeezing, compression, randomness, and multiple parallel AI systems [Sharif et al. 2016; Shokri et al. 2017; Tramèr et al. 2016; Tran et al. 2018; Xu et al. 2018].

7.2 Ethical Dimension

Several institutions and committees are currently developing AI ethical guidelines [Chatila et al. 2017; EU 2019] and standards [ISO 2021], in addition to generic ethical standards for system designs such as IEEE 7000 [IEEE 2021; Widen and Koopman 2022]. Regarding AI-based safety-critical systems, at least two distinct ethical issues must be addressed: *engineering ethics* and *machine ethics* [TUVR 2022].

Engineering ethics is linked to the organization's *safety culture* and associated responsibility and accountability towards the development of such systems [Burton et al. 2020; Dobbe 2022; TUVR 2022]. *Engineering ethics* is also linked to the industry, societal, policymaker and regulatory consensus required to adapt the As Low As Reasonably Practicable (ALARP) principle to these new types of AI-based safety-critical systems that can potentially provide significant societal benefits (e.g., potential car accidents and fatalities reduction with AD systems [Kalra and Paddock 2016; Riedmaier et al. 2020]) with new risks, e.g., which is the acceptable residual risk? [Burton et al. 2020]. Moreover, as analyzed by Widen et al. [Widen and Koopman 2022] and Koopman et al. [Koopman et al. 2021] for the automotive AD domain, the *safety culture* associated to the *engineering ethics* should also encompass the overall business ethics considering aspects such as cooperation with governments for the definition of safe technology regulations, high safety requirements for road testing and deployment, safe management of trade-off dilemmas between financial risks and safety risks, marketing-engineering-regulation coherency for delivered autonomy levels (e.g., L2+ [Cummings and Bauchwitz 2022; Koopman et al. 2021]) and transparency.

On the other hand, *machine ethics* is associated with the moral and ethical decisions that an AI-based *product/runtime* must make during operation. A rich body of research contributions addresses this challenge in the form of dilemma analysis and experiments [Awad et al. 2018; Bonnefon et al. 2016; Burton et al. 2020; Goodall 2014]. In these dilemmas, the autonomous systems are faced with a catastrophic situation where one or several people are in deadly danger in all possible scenarios, and the autonomous system must make a decision that leads to one of these catastrophic scenarios. The key final question is which catastrophic scenario is considered ethically and morally acceptable. For example, in the "moral machine experiment" [Awad et al. 2018], millions of people from different countries provided 40 million decision answers to an autonomous vehicle driving morale dilemma in which people of different ages, genders and professions are in deadly danger. The result of these experiments confirmed that cultural variation and other variation sources (e.g., economic) lead to different moral and ethical decision preferences, concluding that there is no single universal preference for *machine ethics*. However, the German ethical guidelines strictly prohibits decisions made on human classifications (e.g., gender, age) [Koopman et al. 2021; Lütge 2017]. In any case, we should request AI-based safety-critical systems to anticipate and mitigate dangerous situations to avoid such moral dilemmas (e.g., defensive driving strategies in AD system) [Koopman et al. 2021; Lütge 2017].

7.3 Legal Dimension

The European Commission (EU) artificial intelligence act aims to propose a "regulation laying down the set of harmonized rules on artificial intelligence" [EU 2021]. This act establishes that AI-based safety critical systems shall be cataloged as 'high risk' systems subject to specific requirements, such as the conformity assessment process involving notified bodies [EU 2021; TUVR 2022]. That means AI-based safety-critical systems shall be certified/assessed according to applicable domain-specific standards. This is a standardization challenge because for that purpose the industry and standardization committees must first define, update and approve applicable safety standards (see §2.2, §3.3). This also implies detailed technical challenges such as meeting the *auditability* property to support the certification/assessment. Moreover, additional regulations will impose additional specific technical challenges, such as providing *explainability* [McDermid and Jia 2020] to support "the right to obtain an explanation of the decision" made by AI-algorithms ("meaningful information about the logic involved" [EU 2016]) on behalf of an individual, as established by the General Data Protection Regulation (GDPR) [EU 2016].

In addition to this, the legal dimension has also attracted multiple research contributions to address current legal challenges, such as the liability for damages caused by an AI-based *product/runtime* [Burton et al. 2020; Expert Group on Liability and New Technologies 2019; Čerka et al. 2015]. Although the operation of AI-based *products/runtime* is not yet regulated by specific legislation, legal norms require that the offender causing damage must indemnify (liability), or a "person who is responsible for the actions of the offender" [Čerka et al. 2015]. But, for example, if a level 5 autonomous driving system crashes due to decisions made autonomously by the embedded AI technology, in a situation that the manufacturer could not reasonably have foreseen and with no possibility for the passengers to avoid it, who is liable for the accident? Furthermore, "could artificial intelligence become a legal person" with associated offender liability? [Čerka et al. 2015]. The current recommendation of the European Commission [Expert Group on Liability and New Technologies 2019] is that AI not be granted the status of a legal person, as existing parties could instead be held liable in tort for the actions of an AI. However, these and many other related issues remain open multidisciplinary challenges [Burton et al. 2020; Čerka et al. 2015].

Finally, there is also a multidisciplinary collaboration requirement between the legal and engineering dimension. For example, in AD there is a need to translate traffic rules written in human natural language into safety engineering rules for the development and runtime verification of AI-systems [Rizaldi et al. 2017]. This is required for both "holding autonomous vehicles legally accountable" and provide formal safety requirements to reduce the probability of systematic errors [Rizaldi et al. 2017].

8 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This section describes the overall conclusion (§8.1) and future research directions (§8.2).

8.1 Conclusion

This survey summarizes and categorizes a vast and fragmented literature addressing the usage of AI technology for developing safety-critical systems for the industrial and transportation domains, from traditional functional safety to next-generation autonomous systems. Specific AI technology instantiations that perform *automated decision-making* (*A*1) have already been used with compensatory measures (e.g., safety bag) for the development and certification of *automatic* safety-critical systems (e.g., railway interlocking [Klein 1991]). And the use of AI technology for developing specific *heteronomous* safety functions that require human supervision (*A*2) is also common in the latest ADAS systems. However, there is still a significant pending research effort and challenge to define generic AI methods, techniques and processes for developing AI-based safety-critical systems that cannot offload safety management onto humans or non-AI systems. Moreover, there is still a considerable standardization, industrial and research effort remaining to formalize applicable AI-related safety standards, settle best industry practices and define novel technical approaches. There may be a perception that the generic development and certification/assessment of AI-based *autonomous* safety-critical systems (*A*1) will be reached soon. However, we could be at the beginning of the Pareto principle, where 20% of the technological development effort has led to 80% technical results, and AI-based autonomy might seem reasonably achievable soon. However, achieving the following required 20% technical advance might require a considerable additional effort (+80%) due to the difficulty of achieving the required extremely low probability of failure, the necessary systematic capability and providing the supporting evidence as required by present and future safety standards. All in all, we must pave the way toward the development and certification/assessment of AI-based safety-critical systems due to their potential advantages for society and overall industrial interest. So, we expect that the multidisciplinary combination of AI, trustworthiness and safety-critical systems research fields will be an active and vibrant research area for the years to come.

8.2 Future Research Directions

The applicability of AI-technology for developing safety-critical systems leads to multiple, diverse and multidisciplinary challenges. In this Section, we just summarize a set of relevant future research directions aligned with the scope of the survey.

All in all, it is necessary to define an *AI safety engineering* approach with a comprehensive set of generic techniques, life cycles, methods and processes [Jordan 2019; Nordland 2004; Putzer et al. 2021] that could pave the way toward the compliance of AI technology for developing traditional FuSa, heteronomous and autonomous safety-critical systems (*product, runtime, process*). This is an engineering and academia research challenge with two basic types of contributions: "how things can be done" and "how things should be done" [Perez-Cerrolaza et al. 2022; Perez Cerrolaza et al. 2020]. The former refers to the safety adaptation of generic cutting-edge and state-of-the-art AI technology (adapting *Class III* to *Class I-II*). In contrast, the latter refers to a bottom-up development of AI technology natively defined for developing safety-critical systems (*Class I*). And both of them should take into consideration the iterative and dynamic life cycle of AI-based systems (e.g., collect operational data to update the ML model) in the context of industrial and transportation domain systems with long product lifetimes (e.g., >= 30 years [Perez-Cerrolaza et al. 2022]).

As the ML workflow is data-driven, the *data management* must ensure the appropriate *data quality* (e.g., edge/corner cases, data distributional drift) for the safe *model training and verification*. Data must provide a complete, correct and representative specification of the intended safety functionalities, rules and constraints. *Data management* has recurrent challenges and limited research contributions. The systematic error management of *model training* (e.g., AutoML) is also vital for developing safe models, but limited research addresses this challenge. So, both are future research areas with potentially high impact and interest. Not only from a pure AI safety perspective but also from a safety system perspective (e.g., model human driving vs. autonomous driving to better identify representative edge cases and simulation scenarios).

Model verification is an active research area where AI technology is commonly used for the verification *process* of AI-based safety-critical systems (*product*, *runtime*). There are multiple challenges (e.g., test scenarios/case/generation, test classification) and problems to be solved in order to provide technically compliant and economically efficient solutions for the VVT of AI-based safety-critical systems.

System-level safety assurance cases use ML properties to justify that the system is safe for its purpose (e.g., *explainability*, *provability*, *robustness*, *auditability*). So, research contributions that develop AI technology that natively provides these properties, or contributions that extract, measure and verify these properties become crucial. All properties are important, but *explainability* is critical. From a safety engineering perspective, *explainability* is a pivotal attribute in supporting an AI item's understandability, verifiability and auditability. And from a trustworthiness perspective, it is foundational to support the "right to obtain an explanation" and support legal liability analyses providing explainability information for different actors (e.g., engineer, lawyer).

The training tools and platforms on which data is stored, and ML models are trained and verified, are typically based on state-of-the-art solutions with limited or no support for safety systems development (e.g., cloud computing) and non-qualified tools. While academia can provide research contributions, this challenge will likely require an industrial engineering solution.

Additionally, inference execution platforms are an active research area for HPC devices, AI frameworks and middlewares. The avoidance, control and mitigation of random hardware failures and systematic failures, along with the spatial and temporal independence of execution, are common challenges that such execution platforms must address (e.g., diagnostics, temporal predictability). While generic computing devices [Perez-Cerrolaza et al. 2022; Perez Cerrolaza et al. 2020] are already addressing these challenges, specialized devices (e.g., TPU) and AI frameworks still have limited support. Furthermore, there are multiple specialized future research challenges, such as portability and distribution of models among redundant and diverse computing platforms (e.g., FPGA and GPU) [Perez Cerrolaza et al. 2020].

Finally, trustworthiness leads us to multiple, multidimensional and multidisciplinary future research directions combining engineering, law and ethics disciplines, among others. For example, engineering and machine ethics, liability considerations, explainability for different actors, analysis of human vs. autonomous system behaviors.

ACKNOWLEDGMENTS

The findings, conclusions, and opinions expressed herein are solely those of the authors and do not necessarily represent the view(s) of affiliated institutions, organizations or funding bodies. We thank the reviewers for taking the time and effort to review the manuscript and provide valuable comments and suggestions, which helped us improve the manuscript.

REFERENCES

Rusul Abduljabbar, Hussein Dia, Sohani Liyanage, and Saeed Asadi Bagloee. 2019. Applications of Artificial Intelligence in Transport: An Overview. *Sustainability* 11, 1 (2019), 189. https://www.mdpi.com/2071-1050/11/1/189

Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. 2019. Generating Adversarial Driving Scenarios in High-Fidelity Simulators. In *Int. Conf. on Robotics and Automation (ICRA)*. 8271–8277. https://doi.org/10.1109/ICRA.2019.8793740 ISSN: 2577-087X.

Evan Ackerman. 2017. How Drive.ai Is Mastering Autonomous Driving With Deep Learning > Deep learning from the ground up helps Drive's cars handle the challenges of autonomous driving. *IEEE Spectrum* (2017).

A. Adadi and M. Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Prithvi Akella, Ugo Rosolia, Andrew Singletary, and Aaron D Ames. 2020. Formal Verification of Safety Critical Autonomous Systems via Bayesian Optimization. *arXiv preprint arXiv:2009.12909* (2020).

N. Akhtar and A. Mian. 2018. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* 6 (2018), 14410–14430. https://doi.org/10.1109/ACCESS.2018.2807385

Fadi Al-Khoury. 2017. *Safety of Machine Learning Systems in Autonomous Driving*. Thesis.

Mohammad Al-Sharman et al. 2021. A sensorless state estimation for a safety-oriented cyber-physical system in urban driving: Deep learning approach. *IEEE/CAA Journal of Automatica Sinica* 8, 1 (2021), 169–178. https://doi.org/10.1109/JAS.2020.1003474

Berat Mert Albaba and Yildiray Yildiz. 2022. Driver Modeling Through Deep Reinforcement Learning and Behavioral Game Theory. *IEEE Transactions on Control Systems Technology* 30, 2 (2022), 885–892. https://doi.org/10.1109/TCST.2021.3075557

M. Alcon et al. 2020. Timing of Autonomous Driving Software: Problem Analysis and Prospects for Future Solutions. In *IEEE Real-Time and Embedded Technology and Applicat. Symp. (RTAS)*. 267–280. https://doi.org/10.1109/RTAS48715.2020.000-1

Rob Alexander, Hamid Asgari, and Rob Ashmore. 2020. *Safety Assurance Objectives for Autonomous Systems*.

I. Allende, N. M. Guire, J. Perez-Cerrolaza, L. G. Monsalve, J. Petersohn, and R. Obermaisser. 2021. Statistical test coverage for Linux-based next-generation autonomous safety-related systems. *IEEE Access* (2021), 1–1. https://doi.org/10.1109/ACCESS.2021.3100125

Dario Amodei et al. 2016. *Concrete Problems in AI Safety.* Report.

Sara Anastasi, Marianna Madonna, and Luigi Monica. 2021. Implications of embedded artificial intelligence - machine learning on safety of machinery. *Procedia Computer Science* 180 (2021), 338–343. https://doi.org/10.1016/j.procs.2021.01.171

ARP 2023. ARP6983 (WIP) - Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI.

K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38. https://doi.org/10.1109/MSP.2017.2743240

Rob Ashmore, Radu Calinescu, and Colin Paterson. 2021. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *ACM Comput. Surv.* 54, 5 (2021), Article 111. https://doi.org/10.1145/3453444

ASTM 2021. ASTM F3269-21: Standard Practice for Methods to Safely Bound Behavior of Aircraft Systems Containing Complex Functions Using Run-Time Assurance.

J. Athavale et al. 2020b. AI and Reliability Trends in Safety-Critical Autonomous Systems on Ground and Air. In *50th IEEE/IFIP Internat. Conf. on Dependable Syst. and Networks Workshops (DSN-W).* 74–77. https://doi.org/10.1109/DSN-W50199.2020.00024

J. Athavale, A. Baldovin, and M. Paulitsch. 2020a. Trends and Functional Safety Certification Strategies for Advanced Railway Automation Systems. In *IEEE Internat. Rel. Physics Symp. (IRPS).* 1–7. https://doi.org/10.1109/IRPS45951.2020.9129519

AUTOSAR 2022. AUTOSAR (AUTomotive Open System ARchitecture). https://www.autosar.org/. https://www.autosar.org/ Accessed: 2022-08-30.

A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. In *IEEE Trans. on Dependable and Secure Comput.*, Vol. 1. 11–33.

Edmond Awad et al. 2018. The Moral Machine experiment. *Nature* 563, 7729 (2018), 59–64. https://doi.org/10.1038/s41586-018-0637-6

Gerrit Bagschik, Till Menzel, and Markus Maurer. 2018. Ontology based Scene Creation for the Development of Automated Vehicles. In *IEEE Intelligent Vehicles Symposium (IV).* 1813–1820. https://doi.org/10.1109/IVS.2018.8500632 ISSN: 1931-0587.

Baidu. 2021. Apollo CyberRT framework for Autonomous Driving. https://github.com/storypku/CyberRT

Alejandro Barredo Arrieta et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Jenay M. Beer, Arthur D. Fisk, and Wendy A. Rogers. 2014. Toward a framework for levels of robot autonomy in human-robot interaction. *J. Hum.-Robot Interact.* 3, 2 (2014), 74–99. https://doi.org/10.5898/JHRI.3.2.Beer

Halil Beglerovic, Thomas Schloemicher, Steffen Metzner, and Martin Horn. 2018. Deep Learning Applied to Scenario Classification for Lane-Keep-Assist Systems. *Applied Sciences* 8, 12 (2018), 2590.

Vahid Behzadan and William Hsu. 2019. RL-based method for benchmarking the adversarial resilience and robustness of deep reinforcement learning policies. In *Internat. Conf. on Comput. Safety, Rel., and Security.* Springer, 314–325.

Raja Ben Abdessalem et al. 2018. Testing Autonomous Cars for Feature Interaction Failures using Many-Objective Search. In *33rd IEEE/ACM Int. Conf. on Automated Software Eng. (ASE).* 143–154. https://doi.org/10.1145/3238147.3238192

Nelly Bencomo, Jin L.C. Guo, Rachel Harrison, Hans-Martin Heyn, and Tim Menzies. 2022. The Secret to Better AI and Better Software (Is Requirements Engineering). *IEEE Software* 39, 1 (2022), 105–110. https://doi.org/10.1109/MS.2021.3118099

Carl Bergenhem et al. 2015. How to reach complete safety requirement refinement for autonomous vehicles. In *Critical Automotive Applicat.: Robustness & Safety (CARS).*

Christian Berghoff et al. 2020. Towards Auditable AI Systems. In *Auditing AI-Systems: From Basics to Applicat. (Workshop at Fraunhofer Forum).*

C. Bernardeschi, L. Cassano, and A. Domenici. 2015. SRAM-Based FPGA Systems for Safety-Critical Applications: A Survey on Design Standards and Proposed Methodologies. *J. Comput. Sci. Technol.* 30, 2 (2015), 373–390. https://doi.org/10.1007/s11390-015-1530-5

Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning Attacks against Support Vector Machines. In *29th International Coference on International Conference on Machine Learning.* 1467–1474.

Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331.

Alessandro Biondi, Federico Nesti, Giorgiomaria Cicero, Daniel Casini, and Giorgio Buttazzo. 2019. A safe, secure, and predictable software architecture for deep learning in safety-critical systems. *IEEE Embedded Syst. Lett.* 12, 3 (2019), 78–82.

John Birch et al. 2013. *Safety Cases and Their Role in ISO 26262 Functional Safety Assessment.* Vol. 8153. https://doi.org/10.1007/978-3-642-40793-2_15

John Birch, David Blackburn, John Botham, Ibrahim Habli, David Higham, Helen Monkhouse, Gareth Price, Norina Ratiu, and Roger Rivett. 2020. *A Structured Argument for Assuring Safety of the Intended Functionality (SOTIF).* 408–414. https://doi.org/10.1007/978-3-030-55583-2_31

Jp. Blanquart et al. 2004. Software Safety Supervision On-board Autonomous Spacecraft. In *2nd Embedded Real Time Software Congr. (ERTS).*

R. Bloomfield, H. Khlaaf, P. Ryan Conmy, and G. Fletcher. 2019. Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy. *Comput.* 52, 9 (2019), 82–89. https://doi.org/10.1109/MC.2019.2914775

Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. 2016. The social dilemma of autonomous vehicles. *Sci.* 352, 6293 (2016), 1573–1576. https://doi.org/10.1126/science.aaf2654

Markus Borg. 2022. Agility in Software 2.0–Notebook Interfaces and MLOps with Buttresses and Rebars. In *International Conference on Lean and Agile Software Development*. Springer, 3–16.

Markus Borg et al. 2018. Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry. *arXiv e-prints*, Article arXiv:1812.05389 (2018), arXiv:1812.05389 pages.

A. Bosio, P. Bernardi, A. Ruospo, and E. Sanchez. 2019. A Reliability Analysis of a Deep Neural Network. In *IEEE Latin American Test Symp. (LATS)*. 1–6. https://doi.org/10.1109/LATW.2019.8704548

D. A. Bristow, M. Tharayil, and A. G. Alleyne. 2006. A survey of iterative learning control. *IEEE Control Systems Magazine* 26 (2006), 2039–2114.

Fredrik C. Bruhn, Nandinbaatar Tsog, Fabian Kunkel, Oskar Flordal, and Ian Troxel. 2020. Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space. *CEAS Space Journal* (2020). https://doi.org/10.1007/s12567-020-00321-9

P. Burgio et al. 2016. A Software Stack for Next-Generation Automotive Systems on Many-Core Heterogeneous Platforms. In *Euromicro Conf. on Digit. System Des. (DSD)*. 55–59. https://doi.org/10.1109/DSD.2016.84

Simon Burton, Ibrahim Habli, Tom Lawton, John McDermid, Phillip Morgan, and Zoe Porter. 2020. Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective. *Artificial Intell.* 279 (2020), 103201. https://doi.org/10.1016/j.artint.2019.103201

Carmen Cârlan, Barbara Gallina, and Liana Soima. 2021. Safety Case Maintenance: A Systematic Literature Review. In *Computer Safety, Reliability, and Security*, Ibrahim Habli, Mark Sujan, and Friedemann Bitsch (Eds.). Springer International Publishing, 115–129.

Davide Castelvecchi. 2016. Can we open the black box of AI? *Nature* 538, 7623 (2016), 20–23.

Francisco J. Cazorla et al. 2019. Probabilistic Worst-Case Timing Analysis: Taxonomy and Comprehensive Survey. *ACM Comput. Surv.* 52, 1 (2019). https://doi.org/10.1145/3301283

Ján Čegiň. 2020. Machine learning based test data generation for safety-critical software. In *28th ACM Joint Meeting on European Software Eng. Conf. and Symp. on the Foundations of Software Eng.* 1678–1681. https://doi.org/10.1145/3368089.3418538

Ján Čegiň and K. Rástočný. 2020. Test Data Generation for MC/DC Criterion using Reinforcement Learning. In *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 354–357. https://doi.org/10.1109/ICSTW50294.2020.00063

CENELEC. 2020. *CEN-CENELEC Focus Group Report: RoadMap on Artificial Intelligence (AI).* Report. CENELEC.

CENELEC. 2020. EN 50128:2011/A1:2020 - Railway Applications: Communication, signalling and processing systems - Software for railway control and protection systems.

R. Chatila et al. 2017. The IEEE Global Initiative for Ethical Considerations in Artificial Intelligence and Autonomous Systems [Standards]. *IEEE Robotics & Automation Mag.* 24, 1 (2017), 110–110. https://doi.org/10.1109/MRA.2017.2670225

Peter Chemweno, Liliane Pintelon, and Wilm Decre. 2020. Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. *Safety Sci.* 129 (2020), 104832. https://doi.org/10.1016/j.ssci.2020.104832

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2019a. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering.

Baiming Chen, Xiang Chen, Qiong Wu, and Liang Li. 2022. Adversarial Evaluation of Autonomous Vehicles in Lane-Change Scenarios. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2022), 10333–10342. https://doi.org/10.1109/TITS.2021.3091477

Long Chen, Shaobo Lin, Xiankai Lu, Dongpu Cao, Hangbin Wu, Chi Guo, Chun Liu, and Fei-Yue Wang. 2021. Deep Neural Network Based Vehicle and Pedestrian Detection for Autonomous Driving: A Survey. *IEEE Trans. on Intelligent Transportation Syst.* (2021).

Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. 2019b. Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 2 (2019), 292–308. https://doi.org/10.1109/JETCAS.2019.2910232

B. Clough. 2002. Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway. In *Performance Metrics for Intelligent Systems Workshop*.

Darren Cofer et al. 2020. Run-Time Assurance for Learning-Enabled Systems. , 361–368 pages. https://doi.org/10.1007/978-3-030-55754-6_21

Davide Corsi, Enrico Marchesini, Alessandro Farinelli, and Paolo Fiorini. 2020. Formal Verification for Safe Deep Reinforcement Learning in Trajectory Generation. In *4th IEEE Internat. Conf. on Robotic Comput. (IRC)*. IEEE, 352–359.

M. L. Cummings and B. Bauchwitz. 2022. Safety Implications of Variability in Autonomous Driving Assist Alerting. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2022), 12039–12049. https://doi.org/10.1109/TITS.2021.3109555

Werner Dahm. 2010. *Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030.*

William J. Dally, Yatish Turakhia, and Song Han. 2020. Domain-specific hardware accelerators. *Commun. ACM* 63, 7 (2020), 48–57. https://doi.org/10.1145/3361682

O. Daramola et al. 2013. Using Ontologies and Machine Learning for Hazard Identification and Safety Analysis. In *Managing Requirements Knowledge*. Springer Berlin Heidelberg, Berlin, Heidelberg, 117–141. https://doi.org/10.1007/978-3-642-34419-0_6

Sangeeta Dey and Seok-Won Lee. 2021. Multilayered review of safety approaches for machine learning-based systems in the days of AI. *J. of Syst. and Software* 176 (2021), 110941. https://doi.org/10.1016/j.jss.2021.110941

E. T. Dill, S. D. Young, and K. J. Hayhurst. [n. d.]. SAFEGUARD: An assured safety net technology for UAS. In *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. 1–10. https://doi.org/10.1109/DASC.2016.7778009

Roel Dobbe. 2022. System Safety and Artificial Intelligence. In *ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 1584. https://doi.org/10.1145/3531146.3533215

Pedro Domingos. 2018. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.* Basic Books, Inc.

Min Du, Ruoxi Jia, and Dawn Song. 2020. Robust anomaly detection and backdoor attack detection via differential privacy. In *International Conference on Learning Representations (ICLR)*. 1–19.

Peter Du and Katherine Driggs-Campbell. 2019. Finding Diverse Failure Scenarios in Autonomous Systems Using Adaptive Stress Testing. *SAE Int. Journal of Connected and Automated Vehicles* 2, 4 (2019), 241–251. https://doi.org/10.4271/12-02-04-0018 Number: 12-02-04-0018.

EASA. 2021. *EASA Concept Paper: First usable guidance for Level 1 machine learning applications - A deliverable of the EASA AI Roadmap.* Report. European Union Aviation safety Agency (EASA).

Ruediger Ehlers. 2017. Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks. In *Automated Technology for Verification and Analysis (ATVA)*.

Matthis Eicher, Patrick Scharpfenecker, Dieter Ludwig, Felix Friedmann, Florian Netter, and Marius Reuther. 2020. *Process considerations: a reliable AI data labeling process.* Technical Report. Incenda AI and TÜV SÜD.

A. El-Serafy, G. El-Sayed, C. Salama, and A. Wahba. 2015. Enhanced Genetic Algorithm for MC/DC test data generation. In *International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*. 1–8. https://doi.org/10.1109/INISTA.2015.7276794

Meinhard Erben, Wolf Günther, Tobias Sedlmeier, Dieter Lederer, and Klaus-Jürgen Amsler. 2006. Legal Aspects of Safety Designed Software Development, Especially under European Law. In *3rd Eur. Embedded Real Time Softw. (ERTS)*. 6.

EU. 2016. Regulation (EU) 2016/679 of the European parliament and of the council - on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

EU. 2019. *Ethics Guidelines for Trustworthy AI.* Report. European Commission - High-Level Expert Group on Artificial Intell. (HLEG AI).

EU. 2021. Proposal for a regulation of the European parliament and the council - Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts.

Tom Everitt, Gary Lea, and Marcus Hutter. 2018. AGI Safety Literature Review. In *27th Internat. Joint Conf. on Artificial Intell. (IJCAI)*. https://doi.org/10.24963/ijcai.2018/768

Expert Group on Liability and New Technologies. 2019. *Liability for artificial intelligence and other emerging digital technologies.* Report. European Comission.

David D Fan et al. 2020. Bayesian Learning-Based Adaptive Control for Safety Critical Systems. In *IEEE Internat. Conf. on Robotics and Automation (ICRA)*. IEEE, 4093–4099. https://doi.org/10.1109/ICRA40945.2020.9196709

R. Feldt, F. G. de Oliveira Neto, and R. Torkar. 2018. Ways of Applying Artificial Intelligence in Software Engineering. In *IEEE/ACM 6th Internat. Workshop on Realizing Artificial Intell. Synergies in Software Eng. (RAISE)*. 35–41.

Javier Fernandez et al. 2021. Towards Functional Safety Compliance of Matrix-Matrix Multiplication for Machine Learning-based Autonomous Systems. *J. of Syst. Architecture* (2021).

Patrik Feth et al. 2018. Multi-aspect safety engineering for highly automated driving. In *Internat. Conf. on Comput. Safety, Rel., and Security.* Springer, 59–72.

J. F. Fisac et al. 2019. A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems. *IEEE Trans. Automat. Control* 64, 7 (2019), 2737–2752. https://doi.org/10.1109/TAC.2018.2876389

Jörgen Frohm. 2008. *Levels of automation in production systems.* Thesis. https://doi.org/10.13140/RG.2.1.2797.7447

Yarin Gal. 2016. *Uncertainty in Deep Learning.* Ph.D. Dissertation. University of Cambridge.

B. Gangopadhyay et al. 2019. Identification of Test Cases for Automated Driving Systems Using Bayesian Optimization. In *IEEE Intelligent Transportation Systems Conference (ITSC)*. 1961–1967. https://doi.org/10.1109/ITSC.2019.8917103

Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* 16, 1 (2015), 1437–1480.

Florian Geißler et al. 2021. Towards a Safety Case for Hardware Fault Tolerance in Convolutional Neural Networks Using Activation Range Supervision. In *IJCAI Workshop on Artificial Intell. Safety (AISafety)*.

M. Gharib and A. Bondavalli. 2019. On the Evaluation Measures for Machine Learning Algorithms for Safety-Critical Systems. *15th European Dependable Comput. Conf. (EDCC)* (2019), 141–144.

Mohamad Gharib, Tommaso Zoppi, and Andrea Bondavalli. 2021. *Understanding the properness of incorporating machine learning algorithms in safety-critical systems.* Assoc. for Comput. Machinery, 232–234. https://doi.org/10.1145/3412841.3442074

Youcef Gheraibia, Khaoula Djafri, and Habiba Krimou. 2018. Ant colony algorithm for automotive safety integrity level allocation. *Applied Intelligence* 48, 3 (March 2018), 555–569. https://doi.org/10.1007/s10489-017-1000-6

Sangharatna Godboley, Joxan Jaffar, Rasool Maghareh, and Arpita Dutta. 2021. Toward optimal MC/DC test case generation. In *30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 505–516. https://doi.org/10.1145/3460319.3464841

Noah J Goodall. 2014. Ethical decision making during automated vehicle crashes. *Transportation Research Record* 2424, 1 (2014), 58–65.

E. Grade, A. Hayek, and J. Börcsök. 2016. Implementation of a fault-tolerant system using safety-related Xilinx tools conforming to the standard IEC 61508. In *Int. Conf, on Syst. Reliability and Science (ICSRS).* 78–83. https://doi.org/10.1109/ICSRS.2016.7815842

Tuomas Granlund, Vlad Stirbu, and Tommi Mikkonen. 2021. Towards regulatory-compliant MLOps: oravizio's journey from a machine learning experiment to a deployed certified medical product. *SN computer Science* 2, 5 (2021), 1–14.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. *J. of Field Robotics* 37, 3 (2020), 362–386. https://doi.org/10.1002/rob.21918

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5 (2018), Article 93. https://doi.org/10.1145/3236009

Jérémie Guiochet, Mathilde Machin, and Hélène Waeselynck. 2017. Safety-critical advanced robots: A survey. *Robotics and Autonomous Syst.* 94 (2017). https://doi.org/10.1016/j.robot.2017.04.004

G. Hains, A. Jakobsson, and Y. Khmelevsky. 2018. Towards formal methods and software engineering for deep learning: Security, safety and productivity for DL systems development. In *IEEE Int. Syst. Conf. (SysCon).* 1–5. https://doi.org/10.1109/SYSCON.2018.8369576

David J. Hand and Shakeel Khan. 2020. Validating and Verifying AI Systems. *Patterns* 1, 3 (2020), 1–3. https://doi.org/10.1016/j.patter.2020.100037

Fabrice Harel-Canada et al. 2020. Is neuron coverage a meaningful measure for testing deep neural networks?. In *28th ACM Joint Meeting on European Software Eng. Conf. and Symp. on the Foundations of Software Eng.* 851–862.

L. H. Harrison, P. J. Saunders, and P. J. Saraceni. 1993. Artificial intelligence and expert systems for avionics. In *AIAA/IEEE Digital Avionics Syst. Conf.* 167–172. https://doi.org/10.1109/DASC.1993.283552

Hongmei He et al. [n. d.]. The Challenges and Opportunities of Artificial Intelligence in Implementing Trustworthy Robotics and Autonomous Systems. In *3rd Int. Conf. on Intelligent Robotic and Control Engineering*. University of Oxford. https://doi.org/10.1109/IRCE50905.2020.9199244

Philip Helle, Wladimir Schamai, and Carsten Strobel. 2016. Testing of Autonomous Systems - Challenges and Current State-of-the-Art. In *INCOSE Internat. Symp.*, Vol. 26. Wiley Online Library, 571–584. https://doi.org/10.1002/j.2334-5837.2016.00179.x

Jens Henriksson et al. 2018. Automotive safety and machine learning: initial results from a study on how to adapt the ISO 26262 safety standard. In *1st Int. Workshop on Software Eng. for AI in Autonomous Syst.* 47–49. https://doi.org/10.1145/3194085.3194090

Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy, and Cristofer Englund. 2021. Performance analysis of out-of-distribution detection on trained neural networks. *Information and Software Technology* 130 (2021), 106409. https://doi.org/10.1016/j.infsof.2020.106409

H. Hourani, A. Hammad, and M. Lafi. 2019. The Impact of Artificial Intelligence on Software Testing. In *IEEE Jordan Internat. Joint Conf. on Electrical Eng. and Inform. Technology (JEEIT).* 565–570. https://doi.org/10.1109/JEEIT.2019.8717439

Guyue Huang et al. 2021. Machine Learning for Electronic Design Automation: A Survey. *ACM Trans. Des. Autom. Electron. Syst.* 26, 5 (2021). https://doi.org/10.1145/3451179

Xiaowei Huang et al. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Comput. Sci. Rev.* 37 (2020), 100270.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. 2017. Safety Verification of Deep Neural Networks *(Comput. Aided Verification).* Springer Internat. Publishing, 3–29.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated Machine Learning - Methods, Systems, Challenges.* https://doi.org/10.1007/978-3-030-05318-5

IEC 2009. IEC 62267: Railway applications - Automated urban guided transport (AUGT) - Safety requirements.

IEC 2010. IEC 61508(-1/7): Functional safety of electrical/electronic/programmable electronic safety-related systems.

IEC 2014. IEC 62290-1: Railway applications - Urban guided transport management and command/control systems - Part 1: System principles and fundamental concepts.

IEEE. 2021. IEEE 7000: IEEE Standard Model Process for Addressing Ethical Concerns during System Design.

ISO. 2009. ISO 10975: Tractors and machinery for agriculture - Auto-guidance systems for operator-controlled tractors and self-propelled machines - Safety requirements.

ISO. 2011. ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots — Part 1: Robots.

ISO. 2015. ISO 13849-1: Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design.

ISO. 2016. ISO/TS 15066: Robots and robotic devices — Collaborative robots.

ISO. 2017. ISO 16001: Earth-moving machinery - Object detection systems and visibility aids - Performance requirements and tests.

ISO. 2018a. ISO 18497: Agricultural machinery and tractors - Safety of highly automated agricultural machines - Principles for design.

ISO. 2018b. ISO 18758-2: Mining and earth-moving machinery - Rock drill rigs and rock reinforcement rigs - Part 2: Safety requirements.

ISO. 2018c. ISO 25119: Tractors and machinery for agriculture and forestry - Safety-related parts of control systems.

ISO 2018. ISO 26262(-1/11) Road vehicles - Functional safety.

ISO 2019. ISO 17757: Earth-moving machinery and mining — Autonomous and semi-autonomous machine system safety.

ISO. 2019. ISO/PAS 21448: Road vehicles — Safety of the intended functionality (SOTIF).

ISO 2020a. ISO 3691-4: Industrial trucks — Safety requirements and verification — Part 4: Driverless industrial trucks and their systems.

ISO 2020b. ISO/TR 4804 Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation.

ISO 2021a. ISO/AWI TS 5083 Road vehicles — Safety for automated driving systems — Design, verification and validation.

ISO 2021b. ISO/IEC AWI TR 5469: Artificial intelligence — Functional safety and AI systems (draft).

ISO. 2021. ISO/IEC DIS 22989: Information technology - Artificial Intelligence - Artificial intelligence concepts and terminology (draft).

ISO 2021. ISO/IEC DTR 24368 - Information technology - Artificial intelligence — Overview of ethical and societal concerns (Draft).

ISO. 2021. ISO/IEC TR 24030: Information technology — Artificial intelligence (AI) — Use cases.

ISO 2021. ISO/TR 22100-5: Safety of machinery - relationship with ISO 12100 - Part 5: Implications of artificial intelligence machine learning.

Stephen Jacklin et al. 2005. Development of Advanced Verification and Validation Procedures and Tools for the Certification of Learning Systems in Aerospace Applications. In *Infotech@Aerospace*. Amer. Inst. of Aeronautics and Astronautics.

H. Jacobsson. 2005. Rule Extraction from Recurrent Neural Networks: A Taxonomy and Review. *Neural Computation* 17, 6 (2005), 1223–1263. https://doi.org/10.1162/0899766053630350

Henrik Jacobsson. 2006. *Rule extraction from recurrent neural networks.* Thesis.

Georg Jäger, Sebastian Zug, and António Casimiro. 2018. Generic Sensor Failure Modeling for Cooperative Systems. *Sensors* (2018). https://doi.org/10.3390/s18030925

I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder. 2018. Accident Scenario Generation with Recurrent Neural Networks. In *21st International Conference on Intelligent Transportation Systems (ITSC)*. 3340–3345. https://doi.org/10.1109/ITSC.2018.8569661

Eric Jenn et al. 2020. Identifying challenges to the certification of machine learning for safety critical systems. In *10th European Congr. on Embedded Real Time Syst. (ERTS)*. 29–31.

S. Jesenski, J. E. Stellet, F. Schiegg, and J. M. Zöllner. 2019. Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions. In *IEEE Intelligent Vehicles Symposium (IV)*. 502–509. https://doi.org/10.1109/IVS.2019.8813776

Saurabh Jha et al. 2019. ML-Based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 112–124. https://doi.org/10.1109/DSN.2019.00025 ISSN: 1530-0889.

E. N. Johnson, A. J. Calise, and J. E. Corban. 2001. Adaptive guidance and control for autonomous launch vehicles. In *IEEE Aerospace Conf. Proceedings*, Vol. 6. 2669–2682 vol.6. https://doi.org/10.1109/AERO.2001.931288

Michael I. Jordan. 2019. Artificial Intelligence — The Revolution Hasn't Happened Yet. *Harvard Data Science Review* 1 (2019). https://doi.org/10.1162/99608f92.f06c6e61

Norman P. Jouppi et al. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. *SIGARCH Comput. Archit. News* 45, 2 (jun 2017), 1–12. https://doi.org/10.1145/3140659.3080246

Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. 2019. Deep Neural Network Compression for Aircraft Collision Avoidance Systems. *J. of Guidance, Control, and Dynamics* 42, 3 (2019), 598–608. https://doi.org/10.2514/1.G003724

K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer. 2016. Policy compression for aircraft collision avoidance systems. In *IEEE/AIAA 35th Digital Avionics Systems Conf. (DASC)*. 1–10. https://doi.org/10.1109/DASC.2016.7778091

Nidhi Kalra and Susan M. Paddock. 2016. *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* RAND Corporation.

Maryam Kamali, Louise A Dennis, Owen McAree, Michael Fisher, and Sandor M Veres. 2017. Formal verification of autonomous vehicle platooning. *Sci. of Comput. programming* 148 (2017), 88–106.

Guy Katz et al. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Computer Aided Verification*. Springer International Publishing, 97–117.

Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. 2015. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *arXiv:1511.02680* (2015).

Alex Kendall and Roberto Cipolla. 2016. Modelling Uncertainty in Deep Learning for Camera Relocalization. In *IEEE Internat. Conf. on Robotics and Automation (ICRA)*. IEEE, 4762–4769.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. on Intelligent Transportation Syst.* (2021).

Peter Klein. 1991. The safety-bag expert system in the electronic railway interlocking system elektra. *Expert Syst. with Applicat.* 3, 4 (1991), 499–506. https://doi.org/10.1016/0957-4174(91)90175-E

Philip Koopman, Uma Ferrell, Frank Fratrik, and Michael Wagner. 2019. A Safety Standard Approach for Fully Autonomous Vehicles. In *Comput. Safety, Rel., and Security*. Springer Internat. Publishing, 326–332.

Philip Koopman, Benjamin Kuipers, William H. Widen, and Marilyn Wolf. 2021. Ethics, Safety, and Autonomous Vehicles. *Computer* 54 (2021). https://doi.org/10.1109/MC.2021.3108035

Philip Koopman and Michael Wagner. 2016. Challenges in Autonomous Vehicle Testing and Validation. *SAE Int. J. Trans. Safety* 4, 1 (2016), 15–24. https://doi.org/10.4271/2016-01-0128

Philip Koopman and Michael Wagner. 2017. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intelligent Transportation Systems Mag.* 9 (2017), 90–96. https://doi.org/10.1109/MITS.2016.2583491

Philip Koopman and Michael Wagner. 2018. Toward a Framework for Highly Automated Vehicle Safety Validation. In *SAE Tech. Paper 2018-01-1071*. https://doi.org/10.4271/2018-01-1071

R. Krajewski et al. 2018. Data-Driven Maneuver Modeling using Generative Adversarial Networks and Variational Autoencoders for Safety Validation of Highly Automated Vehicles. In *21st Int. Conf. on Intelligent Trans. Systems (ITSC)*. 2383–2390. https://doi.org/10.1109/ITSC.2018.8569971

F. Kruber et al. 2019. Unsupervised and Supervised Learning with the Random Forest Algorithm for Traffic Scenario Clustering and Classification. In *IEEE Intelligent Vehicles Symposium (IV)*. 2463–2470. https://doi.org/10.1109/IVS.2019.8813994

Stefan Kugele, Ana Petrovska, and Ilias Gerostathopoulos. 2021. Towards a Taxonomy of Autonomous Systems. In *15th European Conf. on Software Architecture (ECSA)*.

Lindsey Kuper, Guy Katz, Justin Gottschlich, Kyle Julian, Clark Barrett, and Mykel Kochenderfer. 2018. Toward Scalable Verification for Safety-Critical Deep Networks. *ArXiV* (2018).

Zeshan Kurd, Tim Kelly, and Jim Austin. 2007. Developing artificial neural networks for safety critical systems. *Neural Comput. and Applicat.* 16, 1 (2007), 11–19. https://doi.org/10.1007/s00521-006-0039-9

Zeshan Kurd and Tim P. Kelly. 2004. Using Fuzzy Self-Organising Maps for Safety Critical Systems. In *Comput. Safety, Rel., and Security*, Maritta Heisel, Peter Liggesmeyer, and Stefan Wittmann (Eds.). Springer Berlin Heidelberg, 17–30.

Zeshan Kurd and Tim P. Kelly. 2005. Using safety critical artificial neural networks in gas turbine aero-engine control. In *24th Int. Conf. on Computer Safety, Rel., and Security*. Springer-Verlag, 136–150. https://doi.org/10.1007/11563228_11

Hiroshi Kuwajima, Hirotoshi Yasuoka, and Toshihiro Nakae. 2020. Engineering problems in machine learning systems. *Machine Learning* 109, 5 (2020), 1103–1126. https://doi.org/10.1007/s10994-020-05872-w

Kai Lampka and Adam Lackorzynski. 2019. Using Hypervisor Technology for Safe and Secure Deployment of High-Performance Multicore Platforms in Future Vehicles. In *26th IEEE Int. Conf. on Electronics, Circuits and Syst. (ICECS)*. https://doi.org/10.1109/ICECS46596.2019.8964912

H. Leather and C. Cummins. 2020. Machine Learning in Compilers: Past, Present and Future. In *Forum for Specification and Design Languages (FDL)*. 1–8. https://doi.org/10.1109/FDL50818.2020.9232934

Guanpeng Li et al. 2017. Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Internat. Conf. for High Performance Comput., Networking, Storage and Analysis*. Assoc. for Comput. Machinery. https://doi.org/10.1145/3126908.3126964

Yihao Li, Jianbo Tao, and Franz Wotawa. 2020. Ontology-based test generation for automated and autonomous driving functions. *Information and Software Technology* 117, C (Jan. 2020). https://doi.org/10.1016/j.infsof.2019.106200

R. Lima, A. M. R. da Cruz, and J. Ribeiro. 2020. Artificial Intelligence Applied to Software Testing: A Literature Review. In *15th Iberian Conf. on Inform. Syst. and Technologies (CISTI)*. 1–6. https://doi.org/10.23919/CISTI49556.2020.9141124

Paulo Lisboa. 2001. *Industrial use of safety-related artificial neural networks.* Report. Health & Safety Executive (HSE). http://www.hse.gov.uk/research/crr_pdf/2001/crr01327.pdf

Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26. https://doi.org/10.1016/j.neucom.2016.12.038

Yalin Liu et al. 2020. Unmanned aerial vehicle for internet of everything: Opportunities and challenges. *Computer Communications* 155 (2020), 66–83. https://doi.org/10.1016/j.comcom.2020.03.017

Matt Luckcuck, Marie Farrell, Louise A. Dennis, Clare Dixon, and Michael Fisher. 2019. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. *ACM Comput. Surv.* 52, 5 (2019), Article 100. https://doi.org/10.1145/3342355

Christoph Lütge. 2017. The German Ethics Code for Automated and Connected Driving. *Philosophy & Technology* (2017). https://doi.org/10.1007/s13347-017-0284-0

Yining Ma, Chen Sun, Junyi Chen, Dongpu Cao, and Lu Xiong. 2022. Verification and Validation Methods for Decision-Making and Planning of Automated Vehicles: A Review. *IEEE Transactions on Intelligent Vehicles* (2022), 1–20. https://doi.org/10.1109/TIV.2022.3196396

Y. Ma, Z. Wang, H. Yang, and L. Yang. 2020. Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA J. of Automatica Sinica* 7, 2 (2020), 315–329. https://doi.org/10.1109/JAS.2020.1003021

Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics* 7, 66 (2022), eabm6074. https://doi.org/10.1126/scirobotics.abm6074 arXiv:https://www.science.org/doi/pdf/10.1126/scirobotics.abm6074

Joseph Machrouh, Jean-Paul Blanquart, Philippe Baufreton, J.-L Boulanger, Hervé Delseny, Jean Gassino, Gérard Ladier, Emmanuel Ledinot, Michel Leeman, and Jean-Marc Astruc. 2012. Cross domain comparison of System Assurance. (01 2012).

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Int. Conf. on Learning Representations (ICLR)*.

Klaus Mainzer. 2020. *How Safe Is Artificial Intelligence?* Springer Berlin Heidelberg, Berlin, Heidelberg, 243–266. https://doi.org/10.1007/978-3-662-59717-0_11

I. Martinez et al. 2018. *Safety Certification of Mixed-Criticality Systems.* CRC Press.

Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2022. Software Engineering for AI-Based Systems: A Survey. *ACM Trans. Softw. Eng. Methodol.* 31, 2 (2022). https://doi.org/10.1145/3487043

J. McDermid and Yan Jia. 2020. Safety of Artificial Intelligence: A Collaborative Model. In *AISafety@IJCAI*.

Tim Menzies and Charles Pecheur. 2005. *Verification and Validation and Artificial Intelligence.* Vol. 65. Elsevier, 153–201. https://doi.org/10.1016/S0065-2458(05)65004-8

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations.

S. Mittal and J. S. Vetter. 2016. A Survey of Techniques for Modeling and Improving Reliability of Computing Systems. *IEEE Trans. on Parallel and Distrib. Systems* 27, 4 (2016), 1226–1238. https://doi.org/10.1109/TPDS.2015.2426179

Galen E. Mullins, Paul G. Stankiewicz, R. Chad Hawthorne, and Satyandra K. Gupta. 2018. Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. *Journal of Systems and Software* 137 (March 2018), 197–215. https://doi.org/10.1016/j.jss.2017.10.031

Prabhat Nagarajan et al. 2019. Deterministic Implementations for Reproducibility in Deep Reinforcement Learning. arXiv:cs.AI/1809.05676

Giang Nguyen et al. 2019. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review* 52, 1 (2019), 77–124. https://doi.org/10.1007/s10462-018-09679-z

Odd Nordland. 2004. Can Artificial Intelligence Be Safe?. In *Probabilistic Safety Assessment and Management*, Cornelia Spitzer, Ulrich Schmocker, and Vinh N. Dang (Eds.). Springer London, 400–405.

M. Osborne, H. S. Shin, and A. Tsourdos. 2021. A Review of Safe Online Learning for Nonlinear Control Systems. In *International Conference on Unmanned Aircraft Systems (ICUAS)*. 794–803. https://doi.org/10.1109/ICUAS51884.2021.9476765

M. Ottavi, D. Gizopoulos, and S. Pontarelli. 2018. *Dependable Multicore Architectures at Nanoscale.* Springer. https://doi.org/10.1007/978-3-319-54422-9

Nouara Ouazraoui and Rachid Nait-Said. 2019. An alternative approach to safety integrity level determination: results from a case study. *International Journal of Quality & Reliability Management* 36, 10 (Nov. 2019), 1784–1803. https://doi.org/10.1108/IJQRM-02-2019-0065

Molly O'Brien, William Goble, Greg Hager, and Julia Bukowski. 2020. Dependable Neural Networks for Safety Critical Tasks. In *Engineering Dependable and Secure Machine Learning Systems.* 126–140. https://doi.org/10.1007/978-3-030-62144-5_10

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71. https://doi.org/10.1016/j.neunet.2019.01.012

Joseph N. Pelton and Ram S. Jakhu. 2010. *Space Safety Regulations and Standards.* Butterworth-Heinemann, Oxford. 495 pages. https://doi.org/10.1016/B978-1-85617-752-8.10045-5

A. Pereira and C. Thomas. 2020. Challenges of Machine Learning Applied to Safety-Critical Cyber-Physical Systems. *Machine Learning and Knowledge Extraction* 2, 4 (2020), 579–602.

J. Perez, J. L. Flores, C. Blum, J. Cerquides, and A. Abuin. 2021. Optimization Techniques and Formal Verification for the Software Design of Boolean Algebra Based Safety-Critical Systems. *IEEE Trans. on Ind. Informatics* (2021). https://doi.org/10.1109/TII.2021.3074394

Jon Perez-Cerrolaza, Jaume Abella, Leonidas Kosmidis, Alenjadro J. Calderon, Francisco J. Cazorla, and Jose Luis Flores. 2022. GPU Devices for Safety-critical Systems: A Survey. *ACM Comput. Surv.* (2022). https://doi.org/10.1145/3549526

Jon Perez Cerrolaza, Roman Obermaisser, Jaume Abella, Francisco J. Cazorla, Kim Grüttner, Irune Agirre, Hamidreza Ahmadian, and Imanol Allende. 2020. Multi-core Devices for Safety-critical Systems: A Survey. *ACM Comput. Surv.* 53, 4 (2020). https://doi.org/10.1145/3398665

Chiara Picardi, Colin Paterson, Richard David Hawkins, Radu Calinescu, and Ibrahim Habli. 2020. Assurance argument patterns and processes for machine learning in safety-related systems. In *Workshop on Artificial Intell. Safety (SafeAI)*. 23–30.

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Comput. Surv.* 51, 5 (2018), Article 92. https://doi.org/10.1145/3234150

Luca Pulina and Armando Tacchella. 2012. Challenging SMT solvers to verify neural networks. *AI Commun.* 25, 2 (2012), 117–135.

Laura Pullum et al. 2007. *Guidance for the Verification and Validation of Neural Networks.* John Wiley & Sons, Inc.

Henrik J Putzer et al. 2021. Trustworthy AI-based Systems with VDE-AR-E 2842-61. In *Embedded World.*

Erwin Quiring et al. 2020. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. In *29th USENIX Conference on Security Symposium.*

M. Rabe et al. 2021. Development Methodologies for Safety Critical Machine Learning Applications in the Automotive Domain: A Survey. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*. 129–141. https://doi.org/10.1109/CVPRW53098.2021.00023

Q. M. Rahman, P. Corke, and F. Dayoub. 2021. Run-Time Monitoring of Machine Learning for Robotic Perception: A Survey of Emerging Trends. *IEEE Access* 9 (2021), 20067–20075. https://doi.org/10.1109/ACCESS.2021.3055015

N. Rajabli et al. 2021. Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review. *IEEE Access* 9 (2021), 4797–4819. https://doi.org/10.1109/ACCESS.2020.3048047

Arvind Ramanathan et al. 2016. Integrating symbolic and statistical methods for testing intelligent systems: Applications to machine learning and computer vision. In *Design, Automation & Test in Europe Conf. & Exhibition (DATE)*.

Vincenzo Riccio, Gunel Jahangirova, Andrea Stocco, Nargiz Humbatova, Michael Weiss, and Paolo Tonella. 2020. Testing machine learning based systems: a systematic mapping. *Empirical Software Engineering* 25, 6 (2020), 5193–5254.

S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer. 2020. Survey on Scenario-Based Safety Assessment of Automated Vehicles. *IEEE Access* 8 (2020), 87456–87477. https://doi.org/10.1109/ACCESS.2020.2993730

Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. 2017. *Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL*. https://doi.org/10.1007/978-3-319-66845-1_4

D. Rodríguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja. 2021. Human-Robot Interaction Review: Challenges and Solutions for Modern Industrial Environments. *IEEE Access* 9 (2021), 108557–108578. https://doi.org/10.1109/ACCESS.2021.3099287

Jurgen Ronald. 2013. *Autonomous Driving – A Practical Roadmap (2010-01-2335)*. SAE, 5–26.

RTCA. 2011. DO-178C/EUROCAE ED-12C - Software Considerations in Airborne Systems and Equipment Certification.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intell.* 1, 5 (2019), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Alexander Rudolph, Stefan Voget, and Jürgen Mottok. 2018. A consistent safety case argumentation for artificial intelligence in safety related automotive systems. In *European Congr. Embedded Real-Time Syst. (ERTS)*.

A. Ruospo, A. Bosio, A. Ianne, and E. Sanchez. 2020. Evaluating Convolutional Neural Networks Reliability depending on their Data Representation. In *23rd Euromicro Conf. on Digital System Design (DSD)*. 672–679. https://doi.org/10.1109/DSD51259.2020.00109

SAE. 2010. Aerospace Recommended Practice ARP4754 – Guidelines For Development Of Civil Aircraft and Systems.

SAE. 2014. J3016 - Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.

Aneesa Saeed, Siti Hafizah Ab Hamid, and Mumtaz Begum Mustafa. 2016. The experimental applications of search-based techniques for model-based testing: Taxonomy and systematic literature review. *Applied Soft Computing* 49 (Dec. 2016), 1094–1117. https://doi.org/10.1016/j.asoc.2016.08.030

Rick Salay and Krzysztof Czarnecki. 2018. Using Machine Learning Safely in Automotive Software: An Assessment and Adaption of Software Process Requirements in ISO 26262. *CoRR* abs/1808.01614 (2018).

Rick Salay and Krzysztof Czarnecki. 2019. Improving ML safety with partial specifications. In *Internat. Conf. on Comput. Safety, Rel., and Security*. Springer, 288–300.

R. Salay, R. Queiroz, and K. Czarnecki. 2018. An Analysis of ISO 26262: Machine Learning and Safety in Automotive Software. In *SAE Tech. Paper 2018-01-1075*. https://doi.org/10.4271/2018-01-1075

Mohamed Sallak, Christophe Simon, and Jean-François Aubry. 2006. Evaluating safety integrity level in presence of uncertainty. In *4th Internat. Conf. on Safety and Reliability, (KONBIN)*.

João Alexandre Pedroso Salvado. 2019. *Artificial Intelligence Applied to Software Testing*. Thesis.

F. Fernandes dos Santos et al. 2017. Evaluation and Mitigation of Soft-Errors in Neural Network-Based Object Detection in Three GPU Architectures. In *47th IEEE/IFIP Internat. Conf. on Dependable Syst. and Networks Workshops (DSN-W)*. https://doi.org/10.1109/DSN-W.2017.47

Fernando Fernandes dos Santos, Luigi Carro, and Paolo Rech. 2019. Kernel and layer vulnerability factor to evaluate object detection reliability in GPUs. *IET Comput. & Digital Techniques* 13, 3 (2019), 178–186.

P. Sarathy et al. 2019. Realizing the Promise of Artificial Intelligence for Unmanned Aircraft Systems through Behavior Bounded Assurance. In *IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 1–8. https://doi.org/10.1109/DASC43569.2019.9081649

Sebastian Schirmer et al. [n. d.]. Considerations of Artificial Intelligence Safety Engineering for Unmanned Aircraft *(Computer Safety, Reliability, and Security)*. Springer International Publishing, 465–472.

Volker Schneider. 2021. *Artificial Intelligence and Functional Safety - A summary of the current challenges*. Report. TÜV SUD Rail GmbH. https://metsta.fi/wp-content/uploads/2021/05/Artificial-Intelligence-and-Functional-Safety.pdf

Catherine D. Schuman, Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. 2017. A Survey of Neuromorphic Computing and Neural Networks in Hardware. arXiv:cs.NE/1705.06963

Johann M Ph Schumann and Yan Liu. 2010. *Applications of neural networks in high assurance systems*. Vol. 268. Springer.

Gesina Schwalbe and Martin Schels. 2020. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. In *10th European Congr. on Embedded Real Time Software and Syst. (ERTS)*.

D. Sculley et al. 2015. Hidden Technical Debt in Machine Learning Systems. In *Proc. of the 28th International Conference on Neural Information Processing Systems*. 2503–2511.

D. Serpanos, G. Ferrari, G. Nikolakopoulos, J. Perez, M. Tauber, and S. Van Baelen. 2020. Embedded Artificial Intelligence: The ARTEMIS Vision. *Comput.* 53, 11 (2020), 65–69. https://doi.org/10.1109/MC.2020.3016104

Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. 2018. Uncertainty in Machine Learning: A Safety Perspective on Autonomous Driving. In *Comput. Safety, Rel., and Security*. Springer Internat. Publishing, 458–464.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. Association for Computing Machinery, 1528–1540.

T. Sheridan and W. Verplank. 1978. *Human and Computer Control of Undersea Teleoperators*. Report. MIT.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (SP)*. 3–18. https://doi.org/10.1109/SP.2017.41

Christophe Simon, Walid Mechri, and Guillaume Capizzi. 2019. Assessment of Safety Integrity Level by simulation of Dynamic Bayesian Networks considering test duration. *Journal of Loss Prevention in the Process Industries* 57 (Jan. 2019), 101–113. https://doi.org/10.1016/j.jlp.2018.11.002

SPARC. 2016. *Robotics 2020 - Multi-Annual Roadmap For Robotics in Europe - Horizon 2020 Call ICT-2017 (ICT-25, ICT-27 & ICT-28)*. Report. SPARC (The Partnership for Robotics in Europe).

Chen Sun et al. 2019a. Cross Validation for CNN based Affordance Learning and Control for Autonomous Driving. In *IEEE Intelligent Transportation Systems Conf. (ITSC)*. 1519–1524. https://doi.org/10.1109/ITSC.2019.8917385

Xiaowu Sun, Haitham Khedr, and Yasser Shoukry. 2019b. Formal verification of neural network controlled autonomous systems. In *22nd ACM Internat. Conf. on Hybrid Syst.: Computation and Control*. 147–156.

H. Tabani et al. 2019. Assessing the Adherence of an Industrial Autonomous Driving Framework to ISO 26262 Software Guidelines. In *56th ACM/IEEE Design Automation Conf. (DAC)*. 1–6.

Hamid Tabani, Roger Pujol, Jaume Abella, and Francisco J. Cazorla. 2020. A Cross-Layer Review of Deep Learning Frameworks to Ease Their Optimization and Reuse. In *IEEE 23rd Int. Symp. on Real-Time Distributed Computing (ISORC)*. 144–145. https://doi.org/10.1109/ISORC49007.2020.00030

E. Talpes et al. 2020. Compute Solution for Tesla's Full Self-Driving Computer. *IEEE Micro* 40, 2 (2020), 25–35. https://doi.org/10.1109/MM.2020.2975764

Holger Täubig, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter. 2012. Guaranteeing functional safety: design for provability and computer-aided verification. *Autonomous Robots* 32, 3 (2012), 303–331.

Brian Taylor, Marjorie Darrah, and Christina Moats. 2003a. Verification and validation of neural networks: A sampling of research in progress. *Proceedings of SPIE - The Internat. Soc. for Optical Eng.* 5103 (2003). https://doi.org/10.1117/12.487527

Brian J. Taylor. 2006. *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. Springer Science & Business Media.

Brian J. Taylor, Marjorie A. Darrah, and Christina D. Moats. 2003b. Verification and validation of neural networks: a sampling of research in progress. In *Proc. of SPIE 5103, Intelligent Computing: Theory and Applications*, Vol. 5103. 8–16. https://doi.org/10.1117/12.487527

Francesco Terrosi, Lorenzo Strigini, and Andrea Bondavalli. [n. d.]. Impact of Machine Learning on Safety Monitors. In *Computer Safety, Reliability, and Security*, Mario Trapp, Francesca Saglietti, Marc Spisländer, and Friedemann Bitsch (Eds.). Springer Int. Publishing, 129–143.

N. Theuretzbacher. 1987. ELEKTRA: A System Architecture that Applies New Principles to Electronic Interlocking. *IFAC Proceedings Volumes* 20, 3 (1987), 329–336. https://doi.org/10.1016/S1474-6670(17)55918-2

Stephen Thomas and Dirk Vandenberg. 2019. Harnessing Uncertainty in Autonomous Vehicle Safety. *J. of System Safety* 55, 2 (2019). https://doi.org/10.56094/jss.v55i2.46

Miles S. Thompson. 2008. Testing the Intelligence of Unmanned Autonomous Systems. *ITEA Journal* 29 (2008), 380–387.

Risto Tiusanen, Timo Malm, and Ari Ronkainen. 2020. An overview of current safety requirements for autonomous machines–review of standards. *Open Eng.* 10, 1 (2020), 665–673.

Christoph Torens, Franz Juenger, Sebastian Schirmer, Simon Schopferer, Theresa D. Maienschein, and Johann C. Dauer. 2022. *Machine Learning Verification and Safety for Unmanned Aircraft - A Literature Study*. https://doi.org/10.2514/6.2022-1133

John Törnblom and Simin Nadjm-Tehrani. 2018. Formal verification of random forests in safety-critical applications. In *Internat. Workshop on Formal Techniques for Safety-Critical Syst.* Springer, 55–71.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Conference on Security Symposium*. 601–618.

Brandon Tran, Jerry Li, and Aleksander Mądry. 2018. Spectral Signatures in Backdoor Attacks. Curran Associates Inc., Red Hook, NY, USA, 8011–8021.

Ignacio Trojaola, Iker Elorza, Eloy Irigoyen, Aron Pujana-Arrese, and Carlos Calleja. 2020. *Iterative Learning Control for a Hydraulic Cushion*. 503–512. https://doi.org/10.1007/978-3-030-20055-8_48

C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, and J. Kapinski. 2020. Requirements-Driven Test Generation for Autonomous Vehicles With Machine Learning Components. *IEEE Transactions on Intelligent Vehicles* 5, 2 (2020), 265–280. https://doi.org/10.1109/TIV.2019.2955903

TUVR. 2022. *Basics of Machine Learning with Aspects of Functional Safety and Cybersecurity.* Report. TÜV Rheinland.

ULSE. 2020. UL 4600 - Standard for Evaluation of Autonomous Products.

Rakshith Varadaraju. 2007. *A Survey of Introducing Artificial Intelligence Into the Safety Critical System Software Design Process.* Report. University of Northern Iowa.

K. R. Varshney. 2016. Engineering safety in machine learning. In *Inform. Theory and Applicat. Workshop (ITA).* https://doi.org/10.1109/ITA.2016.7888195

Emil Vassev. 2016. Safe Artificial Intelligence and Formal Methods. In *Leveraging Applicat. of Formal Methods, Verification and Validation: Foundational Techniques*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer Internat. Publishing, 704–713.

Paulius Čerka et al. 2015. Liability for damages caused by artificial intelligence. *Computer Law & Security Review* 31, 3 (2015), 376–389. https://doi.org/10.1016/j.clsr.2015.03.008

VDE. 2021. VDE-AR-E 2842-61: Development and trustworthiness of autonomous/cognitive systems.

W. Wang and D. Zhao. 2018. Extracting Traffic Primitives Directly From Naturalistically Logged Data for Self-Driving Applications. *IEEE Robotics and Automation Letters* 3, 2 (2018), 1223–1229. https://doi.org/10.1109/LRA.2018.2794604

Francis Rhys Ward and Ibrahim Habli. 2020. An Assurance Case Pattern for the Interpretability of Machine Learning in Safety-Critical Systems *(Workshops Comput. Safety, Rel., and Security (SAFECOMP)).* Springer Internat. Publishing, 395–407.

Waymo. 2019. AutoML: Automating the design of machine learning models for autonomous driving. https://blog.waymo.com/2019/07/automl-automating-design-of-machine.html.

L. G. Weiss. 2011. Autonomous robots in the fog of war. *IEEE Spectrum* 48, 8 (2011), 30–57. https://doi.org/10.1109/MSPEC.2011.5960163

William H. Widen and Philip Koopman. 2022. Autonomous Vehicle Regulation & Trust: Impact Of Failures To Comply With Standards. *Journal of Law & Technology (UCLA)* 27, 3 (2022), 169–261. https://doi.org/10.2139/ssrn.3969214

Nan Wu and Yuan Xie. 2022. A Survey of Machine Learning for Computer Architecture and Systems. *ACM Comput. Surv.* 55, 3 (2022), Article 54. https://doi.org/10.1145/3494523

Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Network and Distributed System Security (NDSS) Symposium.*

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. 2020. ML-LOO: Detecting Adversarial Examples with Feature Attribution. AAAI Press, 6639–6647.

Junko Yoshida. 2020. *Unveiled: BMW's Scalable AV Architecture.* IEEE.

Katsuba Yurii and Grigorieva Liudmila. 2017. Application of Artificial Neural Networks in Vehicles' Design Self-Diagnostic Systems for Safety Reasons. *Transportation Research Procedia* 20 (2017), 283–287. https://doi.org/10.1016/j.trpro.2017.01.024

Jin Zhang and Jingyue Li. 2020. Testing and verification of neural-network-based safety-critical control software: A systematic literature review. *Inform. and Software Technology* 123 (2020), 106296. https://doi.org/10.1016/j.infsof.2020.106296

Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* (2020).

Ding Zhao, Xianan Huang, Huei Peng, Henry Lam, and David J. LeBlanc. 2018. Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (March 2018), 733–744. https://doi.org/10.1109/TITS.2017.2701846

Q. Zhu et al. 2021. Safety-Assured Design and Adaptation of Learning-Enabled Autonomous Systems. In *26th Asia and South Pacific Design Automation Conference (ASP-DAC).* 753–760.