The Journal of Systems & Software 176 (2021) 110941

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Multilayered review of safety approaches for machine learning-based systems in the days of AI☆

Sangeeta Dey [a], Seok-Won Lee [a,b,*]

[a] Department of Artificial Intelligence, Ajou University, Suwon City, 16499, Republic of Korea
[b] Department of Software and Computer Engineering, Ajou University, Suwon City, 16499, Republic of Korea

ARTICLE INFO

Article history:
Received 12 July 2020
Received in revised form 2 December 2020
Accepted 17 February 2021
Available online 6 March 2021

Keywords:
Autonomous systems
Intelligent software systems
Machine learning
Safety analysis
Software engineering

ABSTRACT

The unprecedented advancement of artificial intelligence (AI) in recent years has altered our perspectives on software engineering and systems engineering as a whole. Nowadays, software-intensive intelligent systems rely more on a learning model than thousands of lines of codes. Such alteration has led to new research challenges in the engineering process that can ensure the safe and beneficial behavior of AI systems. This paper presents a literature survey of the significant efforts made in the last fifteen years to foster safety in complex intelligent systems. This survey covers relevant aspects of AI safety research including safety requirements engineering, safety-driven design at both system and machine learning (ML) component level, validation and verification from the perspective of software and system engineers. We categorize these research efforts based on a three-layered conceptual framework for developing and maintaining AI systems. We also perform a gap analysis to emphasize the open research challenges in ensuring safe AI. Finally, we conclude the paper by providing future research directions and a road map for AI safety.

1. Introduction

Recently, artificial intelligence (AI) has received increased attention from many sectors and fields. These diverse areas include autonomous driving, computer vision, medical diagnosis systems, gaming systems, etc. Various industries are overwhelmed by the positive potential of AI. This unprecedented advancement of AI has altered our perspectives on various business functions, including software-intensive intelligent systems development process. Machine learning (ML) is one of the greatest contributors to this revolution. The task of a software engineer has now transformed from writing thousands of lines of code to training, retraining, testing, and maintaining a learning model. However, it is essential to realize that every advancement of technology comes with surprises and concerns, and AI is no exception. AI revolution is not at its inception any more. However, the means to ensuring safety, transparency, and level of fidelity of AI systems are still unclear. Many researchers have already expressed apprehension related to the advent of AI especially in high-risk environments (Agrell et al., 2018; Russell et al., 2015). Moreover, AI systems that equal or even exceed humans in cognitive tasks are both appealing and alarming (Everitt et al., 2018; Brundage, 2015; Soares and Fallenstein, 2014). Introducing a sense of ethics and morality has long been discussed as the top priority for the days of AI (Davis, 2015; Winfield et al., 2014). Therefore, amid the excitement about improved efficiency due to AI, it is worth engaging in discussions on its potential risks, new challenges, and questionable impact on safety issues (Amodei et al., 2016).

The process of reduction of human interventions in many sectors of industries is already underway. Automotive, aerospace etc. are heavily using ML algorithms to paving their way towards increasing autonomy (Monkhouse et al., 2017). A fully automated driver-less car is not a far-fetched dream in the days of AI (Koopman and Wagner, 2018). However, a few recent accidents involving semi-autonomous cars (Greenblatt, 2016) have negatively impacted our trust in full autonomy. It has been argued that the increasing autonomy of AI systems can have enormous impact on humanity (Amodei et al., 2016; Monkhouse et al., 2017). Use of systems that rely heavily on the decision making of an ML model should be a tremendous concern in high-risk environment. Although introduction of ML in controlling complex engineering systems has been proved to be more efficient than unpredictable erroneous human controls in many cases, the robustness of such ML models should be assessed.

However, as stakeholders from various backgrounds and expertise participate in the process of ensuring safety of a complex systems, a consensus on the fundamental properties of autonomous systems; e.g. controllability, explainability, robustness

☆ Editor: Burak Turhan.

* Corresponding author at: Department of Software and Computer Engineering, Ajou University, Suwon City, 16499, Republic of Korea.
E-mail addresses: deysangeeta@ajou.ac.kr (S. Dey), leesw@ajou.ac.kr (S.-W. Lee).

https://doi.org/10.1016/j.jss.2021.110941
0164-1212/

to uncertain environment, etc. is necessary (Mikolov et al., 2016; Koopman, 2018b). Functional safety experts worry about the gap between the vision of the current automotive industry and the scope of the current safety standards landscape (Monkhouse et al., 2017). Traditional risk model and safety analysis (Stringfellow et al., 2010a) are also inadequate to handle the immense use of AI in safety-critical systems. Moreover, how the activities performed by experts of diverse expertise integrate to foster an emergent property like safety is not clear. For instance, how the artifacts flow across the various layers of methodical systems engineering at different levels of abstractions is still an open question. It is still unclear how the traceability can be maintained in a collaborative environment where the stakeholders not necessarily speak the same terms. Singla et al. have analyzed the differences between ML-based agile software projects and traditional agile software projects in terms of their execution processes, issues faced and the terms used to describe same concepts in these two types of workload (Singla et al., 2018).

Therefore, we believe that, we need to analyze the state-of-the-art safety approaches through the lens of a methodological engineering process. To the best of our knowledge, no systematic literature review of AI safety from a software engineering perspective has been done in recent times. The landscape of AI safety and beneficence research (Mallah, 2017) discusses a vast range of research topics directed to AI safety. This article includes research areas such as validation, verification, control, security, etc. However, in order to perform a gap analysis on the state-of-the-art AI safety approaches, it is imperative to have a bigger picture of the recent research efforts from an engineering process perspective.

Research questions that we try to answer through our research are:

RQ1: How can we easily comprehend the complexity and challenges involved in fostering safety of a complex intelligent systems?

Rationale: AI systems today are very complex at many layers. We argue that, not only the technical advances force us to give a deeper thought to safety related issues, but also the diversity of stakeholders participating in the systems engineering process amplifies the complexity. For example, a system level safety engineer may not be fully aware of the impact of the minute loopholes in the software module which vastly relies on ML algorithms. As different stakeholders with diverse expertise work at different levels of abstractions in the process of fostering safety, it is important to visualize the state-of-the art safety approaches from a layered perspective.

RQ2: How have safety concerns been addressed by the researchers along the phases of SE process?

Rationale: We acknowledge the paradigm shift that has taken place in the software development process. However, we believe that, the foundation of software engineering (SE) process entailing phases such as requirements engineering (RE), design, development, validation, verification and maintenance, still provides a strong methodological foundation to the whole process of intelligent software engineering even in this new paradigm. Understanding data requirements, designing the parameters and features of a learning model, training, tuning and testing of the ML model, and finally, maintaining or updating the model over time are new additions to the activities of phases such as RE, design, development, V & V, maintenance respectively. In our study we would like to map the research efforts in the field of AI safety along the phases of SE.

RQ3: What are the gaps in the current research efforts?

Rationale: After we find the answers to RQ1 and RQ2, we can have an accurate visualization of the areas that have not been explored yet. Finding the gap can eventually help us provide possible directions for future research.

RQ4: What are the future directions that may help reduce the gaps?

Rationale: We further want to analyze the existing gaps and provide our preliminary vision on the ways to reduce those gaps. Our paper finds the scope of improved collaborations among diverse stakeholders while analyzing current state-of-the-art safety approaches from a multidisciplinary engineering process perspective.

In summary, the major contributions of this paper are:

– We provide a conceptual three-layered framework that helps us visualize the inherent complexity in developing and maintaining complex intelligent systems involving various stakeholders.

– We identify the challenges and risks that are of major concern along each layer of the framework.

– We review significant work conducted to address safety-related issues in intelligent systems in the last fifteen years.

– We perform a gap analysis to identify what aspects we are missing in literature and practice.

– Finally, we provide a research road map to expand the ongoing research on safe AI.

The remainder of the paper is organized as follows. In Section 2 we discuss the background of the research on safe AI systems. We summarize the related surveys in this area and explain the position of our survey in Section 3. A detailed research method of conducting a literature review is explained in Section 4. Section 5 introduces the proposed three-layered conceptual framework to better visualize the engineering process of complex intelligent systems. It also elaborates on the challenges and significant research efforts address the AI safety-related problems along each layer of the framework. We discuss our findings of literature review based on the research questions in Section 6. The limitation and threats to validity of this study are discussed in Section 7. Finally, we conclude the paper in Section 8.

2. Background

2.1. What is a “Safe AI system”?

Before we investigate the formal explanation of a safe AI system, we discuss the definition of ‘safety’. The concept of safety has long been defined and analyzed from multiple perspectives. Various domains have explained safety in different ways. Leveson explained that safety means absolutely no harm to people and no accidents with or without harming people (Leveson, 2016). According to IEC 61508, “Safety is freedom from unacceptable risk”. This definition leads us to the concept of ‘Risk’ (ISO/IEC, 2010a). To analyze safety, the level of risk should be assessed. This definition also seems to be more practical than the previous one as it is almost impossible to engineer a fault-free system. Rather, this definition leaves an existing scope of risk up to a certain level of tolerance. Another definition of safety given by ISO 26262 (ISO, 2018) is “safety means the absence of unreasonable risk”. This definition involves not only the probabilistic analysis of risk, but also the mechanisms to avoid harm and the possibilities of occurrences of such situations.

Recently, with the unprecedented emergence of AI based systems, the concept of ‘safety’ has started to entail a broader concept than the discussed definitions. AI techniques come with various inherent risks and behavior uncertainties due to a vast range of reasons: data-driven behavior (instead of code or rule-driven), self-learning or exploration, black-box nature, etc. As these systems are mostly driven by a few system-level objectives specified by the designers, it is important to ensure that the systems will not try to achieve their goals in an undesirable manner. In other words, the current safety concerns are not limited to failure, non-availability, and wrong outcomes; rather the concerns

include reward hacking, negative side-effects, unsafe exploration, and insufficient robustness to distributional shift, etc. (Amodei et al., 2016). As argued by Varshney (2016), it is not enough to minimize risk by defining a risk-minimal loss function. Instead, it is also important to address epistemic uncertainty in the underlying distribution of training and test instances. Moreover, as these AI systems are deployed in a socio-technical environment, the perception of safety or 'safe system' depends considerably on the confidence of society in such systems. From this perspective, a safe AI system should not only be engineered to behave safely but also be capable of explaining its behavior to a wide range of society.

2.2. When is a system "safe enough"?

While nowadays we have a common understanding that no system can be absolutely fail-free, we still do not have a consensus on the safety measurement criteria. We have various safety standards to certify a system as safe enough to deploy. However, the overall quantification (formalism) of how much safe is safe enough vary widely across various domains. Littlewood made a significant contribution in quantifying software safety with the help of evidence-based arguments and the confidence in those arguments (Littlewood and Wright, 2007). With the help of multi-legged arguments, he showed how diversity in arguments can be utilized to prove dependability of a software system with a certain confidence. For example: if the overall argument is to support a claim of 10^{-4} pfd (probability of failure on demand), a statistical leg can support this claim with direct evidence whereas a logical leg can also fine-tune its confidence in the claim based on the confidence in the statistical leg.

Automotive industries that produce AI-based autonomous vehicles have also faced with the same challenge to acquire sufficient confidence in the safe behavior of a newly designed vehicle. As discussed earlier, with the advent of data-driven learning techniques, there is little scope of finding bugs in the code or models. Apart from gaining confidence in the usage of right set of training data, one of the promising ways to ensure safety is to test the system thoroughly in simulation-based or real-world situations. However, the challenge is to make sure that the designers have explored all possible scenarios and surprises that the system can encounter during testing. As Koopman mentioned in Koopman (2018b), miles on road is regarded as evidence of the safe behavior of the vehicle without any clear explanation of how many miles are enough to gain a certain level of confidence in the safety claim. For instance, if we assume there are 100 surprises in total, each arriving in every 100 million miles, then to prove that the systems have encountered all the surprises and has been corrected to handle each of those surprises, it has to be tested on at least one or two billion miles. This is nearly unattainable. Therefore, many researchers are now focusing on designing frameworks to plan to test wisely while limiting expensive on-road testing. We discuss more on the testing and statistical evidences of AI systems safety in Section 5.3.

3. Related surveys and our position

Since safety concerns of AI systems have gained attention in the last few years, many literature reviews, technical reports, and surveys have consolidated the relevant state-of-the-art approaches. We collected the related surveys while searching for the primary studies.

1. Based on the search strings (discussed in detail in the next section), we first collected the secondary studies (literature surveys, questionnaires, reports, summary, case studies, etc.). We found more than 40 such studies that matched with the relevant search keywords.

2. We selected only 14 studies based on their relevance to safety, perspective, and coverage. The selection criteria were as follow:

(i) The study is published between 2015–2020.

(ii) The study is peer-reviewed and written in English.

(iii) The study is focused on the safety aspects of AI, ML, intelligent software engineering, safety-critical systems (that use ML heavily).

The studies that review the application of various ML techniques to automate software engineering or other domain (without covering the safety aspect) have been excluded. We have deliberately included three recent studies on the integration of the software engineering process and ML life-cycle as part of our related work (despite violating the third criteria) (Ishikawa and Yoshioka, 2019; Serban et al., 2020; Lwakatare et al., 2020; Wan et al., 2019). These studies do not solely focus on the safety aspects of intelligent software. However, we include them as these studies solidify our view of performing a literature survey on safety approaches from a software engineering perspective.

We can generally categorize the selected secondary studies as — domain-based (concerning a particular domain such as automobile, robotics, etc.), AI algorithm/ ML techniques-based (focusing on a particular type of algorithm or ML technique like deep learning, reinforcement learning), general AI systems-based(surveying a vast range of safety-related research efforts applicable to all AI systems in general), and engineering process-based (surveys the safety approaches from a systematic engineering process point-of-view that considers collective contributions from diverse stakeholders at each phase).

There are a few domain-specific surveys (Luckcuck et al., 2019; Yurtsever et al., 2020; Guiochet et al., 2017; Guo et al., 2019; Nascimento et al., 2019) in the field of safety-critical systems such as driverless cars and robotics, that cover a vast range of engineering aspects such as design, control, architecture, etc. However, it is either difficult to understand the safety concerns of AI algorithms or ML techniques used in these systems (Luckcuck et al., 2019; Yurtsever et al., 2020; Guiochet et al., 2017) or it is difficult to generalize the safety concerns for all safety-critical domains (Guo et al., 2019; Nascimento et al., 2019). Numerous reviews have been published on the robustness of ML techniques (Ozdag, 2018; Chakraborty et al., 2018; Xu et al., 2020; Garcıa and Fernández, 2015) and assurance of ML-based systems (Xiang et al., 2018; Borg et al., 2018; Schumann et al., 2010; Zhang et al., 2020). The target readers of those surveys are expected to have deep technical knowledge in ML/AI algorithms to comprehend such studies as they entail many minute technical details. There are also a few technical reports, research agendas published by Stanford AI research groups (Dewey et al., 2015; Mallah, 2017). These reports cover a broad range of topics related to AI safety, security, privacy, fairness, trust, etc. While these studies provide an overview of the scope of research contributions in the field of safe AI, analyzing the current research efforts from an engineering process perspective is not in the scope of those studies.

Our paper differs from the other three types of surveys mentioned above in various ways. The objective of this paper is not to discuss the detailed technicality of state-of-the-art safety approaches. Rather our goal is to explore how research efforts fit from a layered systems engineering perspective. Recently, few researchers and practitioners have conducted literature reviews, questionnaires, surveys to understand the challenges of engineering ML-based software (Ishikawa and Yoshioka, 2019; Serban et al., 2020; Lwakatare et al., 2020; Wan et al., 2019). Wan et al. and Ishikawa et al. have conducted questionnaire surveys including experienced ML-based software engineers to summarize the

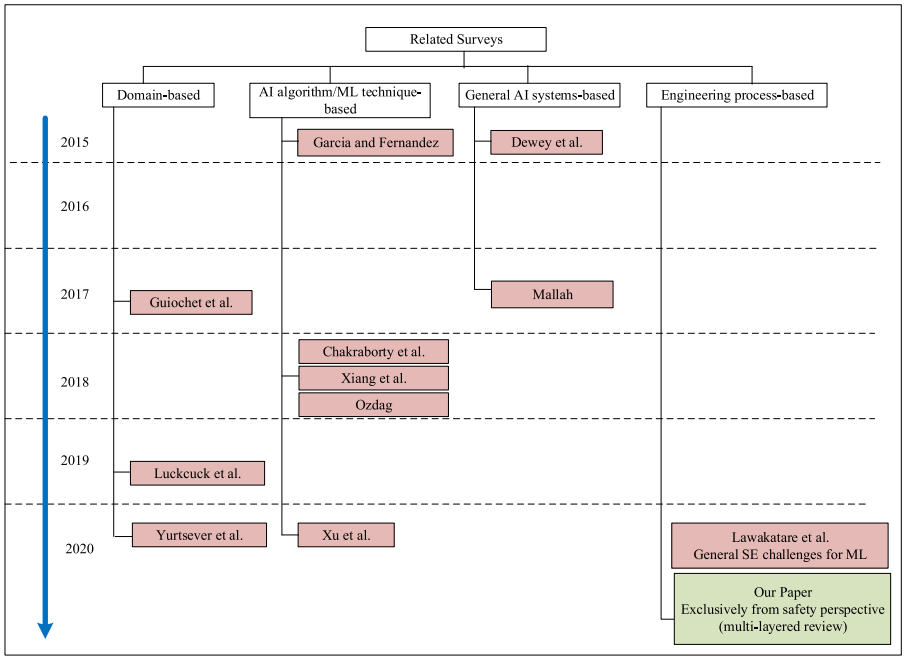

Fig. 1. Related surveys and out position.

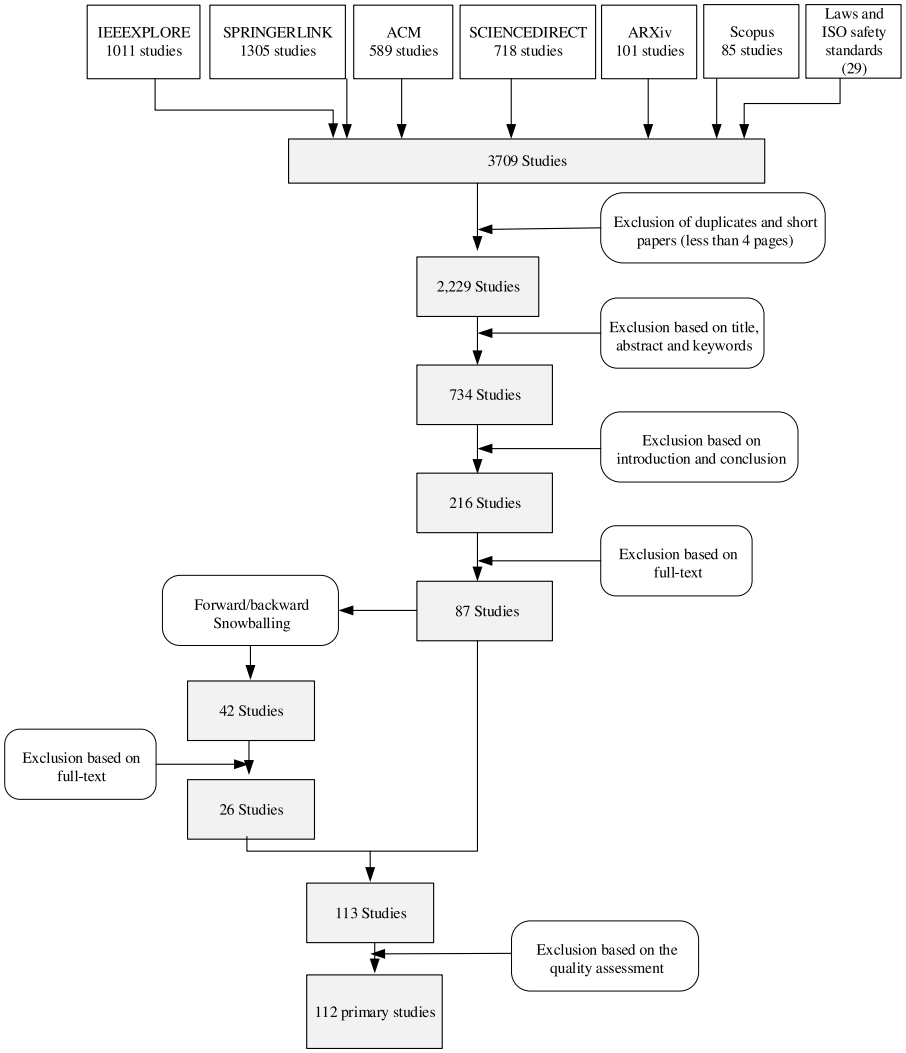

Fig. 2. Study selection process.

Table 1
Search strings used to retrieve relevant primary studies.

Objective	Search String
To retrieve studies on Artificial Intelligence and safety or trust or reliability or assurance-related concerns and corresponding solutions	"artificial intelligence" AND (safe OR trust OR reliab OR assurance)
To retrieve studies on safety, reliability, risks or robustness or assurance-related concerns of machine learning or deep learning and corresponding solutions	("machine learning" OR "deep learning") AND (safe OR robust OR risk OR reliability OR assurance)
To retrieve studies on automotive industries or any autonomous/unmanned safety-critical systems and safety-related issues.	(auto OR unmanned OR safety-critical) AND (safe OR verification)
To retrieve studies on machine learning (especially neural networks)-based systems verification or testing	("machine learning" OR "neural networks" OR "deep learning") AND (verification OR testing)
To retrieve studies that focus on the correlation between software engineering, requirements engineering, and machine learning.	("software engineering" OR "requirements")AND "machine learning"

software engineering challenges and corresponding sources of difficulties faced during the development and validation of such software (Ishikawa and Yoshioka, 2019; Wan et al., 2019). Serban et al. have conducted a survey including ML practitioners to analyze how software engineering best practices are being used by the ML teams and how effective those practices are in the case of ML-based systems (Serban et al., 2020). Silverio et al. have discussed challenges of trustworthy AI-based autonomous systems in industrial settings and also have provided future directions on reducing the gap between the development and operation of such systems (Martínez-Fernández et al., 2020). Lwakatare et al. have summarized 23 software challenges and 8 solutions related to the large-scale ML systems after conducting a literature survey (Lwakatare et al., 2020). In quite a similar direction, Anh and Pekka have discussed the challenges and have provided a research agenda on continuous experimentation of AI software in large-scale systems (Nguyen-Duc and Abrahamsson, 2020). Few of the challenges discussed in our survey overlap with their findings. However, their studies do not focus on safety. Instead that study focus on the quality attributes of software in general.

To the best of our knowledge, no survey has been done yet to analyze the relevant research areas through the lens for diverse stakeholders and their contribution to the engineering process from a safety perspective. Our paper aims to find the scope of collaboration to fill the gaps in the continuity of the recent research efforts when analyzed from a multidisciplinary engineering process perspective. Fig. 1 summarizes the relevant literature surveys and our position.

4. Research method

4.1. Research questions

We have already explained the research questions in Section 1. RQ1 is slightly different than the rest of the questions as we try to answer RQ1 based on the overall view of how a complex AI system is conceived, designed, developed, verified, and maintained. We envision a three-layered conceptual framework that facilitates further analysis of the recent research efforts along the layers. We describe our vision in detail in Section 5 and explore the state-of-the-art safety approaches based on our conceptualization of the problem space and solution space regarding safe AI.

4.2. Search strings used

We conducted the search on 29-Oct-2020 with the search strings to retrieve the recent studies as mentioned in Table 1. We initially attempted to search based on the title and abstract. However, we were overwhelmed by more than 10,000 spurious search results for most of the queries. Therefore, we decided to limit our initial search on the titles and follow snowballing method afterward to retrieve other relevant studies that were missed out by the search operation. The search results were filtered by the range of publication years from 2005 to 2020. In case of Springer, the search results were further filtered by discipline and sub-discipline. Detailed review protocol, list of search strings used for each database, summary of primary and secondary studies can be found in the replication package. (Dey and Lee, 2021)

4.3. Search strategies and data sources

We conducted the search in two phases. In Phase-1, we directly searched databases like IEEE xplore, ScienceDirect, Springer-link, ACM Library, Scopus and arXiv. We collected research papers published within the 15-year time period between 2005 and 2020. After getting a moderate number of search results from these data sources, we followed the forward and backward snowballing approach (Wohlin, 2014) to find new relevant papers (from the references and the citation lists) in Phase-2.

4.4. Study selection process

In Phase-1, we initially found more than 3000 papers directly by performing database search. However, not all of those studies were relevant to our literature review. Our main aim was to collect studies focusing on the intersection of safety engineering, artificial intelligence, and software engineering. Therefore, for example, studies focusing only on safety engineering without considering autonomy, AI techniques, or SE process were excluded from the collection. The detailed inclusion and exclusion criteria are described in Table 2. In Phase-1, we removed duplicates and excluded irrelevant papers after reading the title, abstract, and in some cases introduction. In Phase-2, we read the full-text to thoroughly understand the motivation and contribution of each work. Finally, we selected only 112 papers that we agreed were relevant and significant for our study (Fig. 2).

4.5. Study quality assessment

We assessed the quality of the selected studies based on a set of six questions. A complete list of the questions is provided in Table 3. Most of the questions can have three answers: Yes (Y) = 1, No (N) = 0, Partial (p) = 0.5 except for Q3, Q4 and Q6. Q3 can only have two answers Y = 1 for general and N = 0 for a

Table 2
Overview of the inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
1. Primary studies or safety standards. 2. Studies published in the English language. 3. Peer-reviewed studies except for the studies published on Arxiv. 4. If the study is published on Arxiv then it should have at least one citation by a peer-reviewed paper. 5. Studies that focus on AI, ML intelligent Software Engineering and corresponding safety-related issues. 6. Studies published between 2005 and 2020.	1. Secondary studies. 2. Duplicate studies. (only longer and more complete versions were accepted.) 3. Studies that only focus on safety-related research (safety lifecycle, safety artifacts, etc.) without consideration of the recent advancement of AI. 4. Short studies (less than 4 pages).

Table 3
Quality assessment criteria.

QA scenario component	Security requirements components
Are the motivation and goal of the study well explained in the paper?	Y = 1, N = 0, p = 0.5
Is the proposed approach adequately explained with necessary details?	Y = 1, N = 0, p = 0.5
Is the study applicable to all domains or to a specific domain?	Y = 1, N = 0
Does the study provide value for research or practice?	Y=1, N=0
Is there any discussion about the results or threats to validity?	Y=1, N=0, p=0.5
Does the study has a section on related work?	Y=1, N=0
Does the study provide future direction or open scope for further research or investigation?	Y=1, N=0, p=0.5

domain-specific solution. Similarly, Q4 has two answers Y = 1 for contribution to research and N = 0 for contribution to practice. In general, the papers that scored above 4 out of 7, were selected. Each of the selected papers was thoroughly assessed by each of the authors individually and any disagreement was discussed till a consensus was reached.

Except for one paper that was published in IEEE software magazine, all other papers qualified to be part of our study. The reason we excluded that 4 page long paper on "Software engineering for machine learning applications" was it lacked detailed explanation of the solution and discussion on the limitation as it was only a theme issue. We made another exception regarding the quality assessment in case of safety standards. Most of the articles discussing standards are from web-based search or part of technical reports, etc. These articles did not go through the rigorous quality assessment process as per the criteria mentioned in Table 3. Instead, they have been selected based on their timeliness and relevance to our study.

5. Three-layered conceptual framework for safety-driven AI system engineering

As appropriately argued by Koopman, many different areas require coordination to ensure safety (Koopman and Wagner, 2017). Acknowledging the fact that there is a cognition gap among the various disciplines involved in the process of risk analysis, we need to provide a set of guidelines to the diverse stakeholders. To formalize the participation and responsibility of each stakeholder, we propose a three-layered framework to conceptualize complex intelligent systems (Fig. 3). The framework shows how a safe output space is achieved for such systems starting from an initial incomplete problem space. We can visualize how diverse the stakeholders and the primary artifacts are, in each layer at a system level and ML-based component level. In the days of AI, at the ML-based component level, the stakeholders come from AI or ML specific background. This was not very common in the case of traditional systems without any machine intelligence. These paradigm shifts bring in a lot of challenges in exploring the problem space and solution space. The layers of the framework are listed below:

- **Problem definition layer:**
  In this layer, at a system level, domain experts and requirements engineers work closely with users to gather requirements. The aim of these participants is to understand the expectations of the users and the society, goals of the system, possible situations/scenarios, etc. With the introduction of ML-based components, conceiving a rich problem space has become more complex than in traditional systems. ML experts, data scientists, HCI experts need to work closely with system-level stakeholders to understand the ML model requirements, data requirements, domain definitions, quantitative targets, etc.
- **Safety-driven modeling and development layer:**
  In this layer system engineers and safety experts work in collaboration to perform system-level risk analysis, risk handling of emergent behavior of the system. In other words, in this layer risk-minimizing safety goals, requirements, etc. are derived from system-level risks. Safety standards are followed throughout the process of designing and developing the system. At the component level, inherent risks of ML models, effects of adversarial attacks, corner cases are identified. Unlike traditional software-intensive systems, ML-based components are designed and trained based on objective functions following AI/ML specific safety standards. Safe learning decisions on the safety-aware rewards, penalty, etc. are made by the ML experts at this layer.

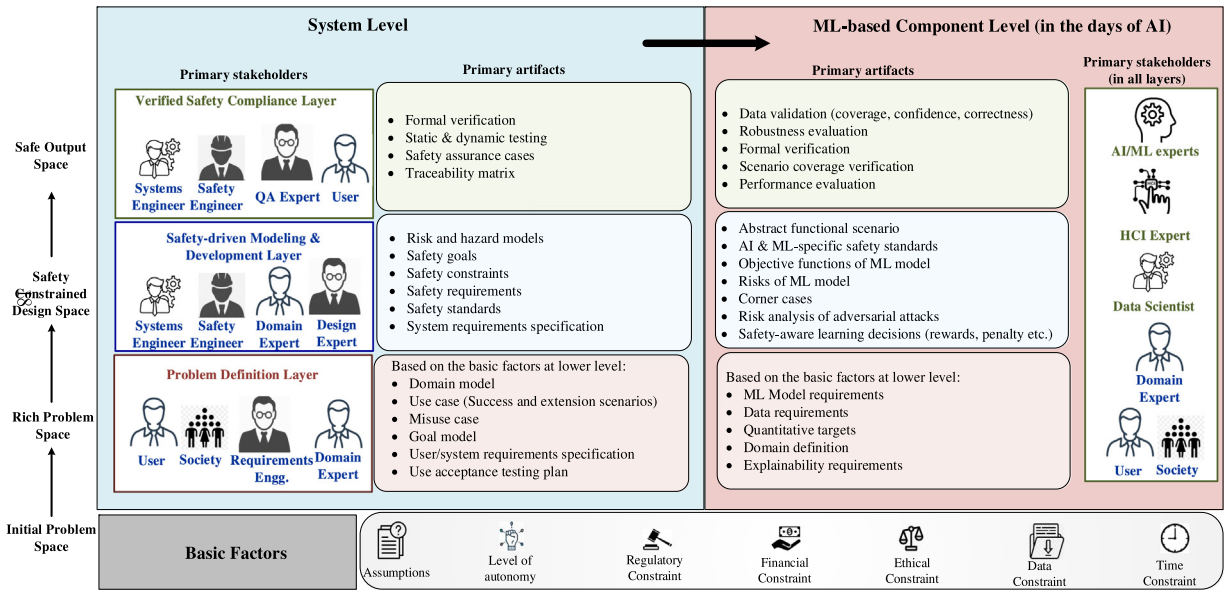

Fig. 3. An integrated three-layered conceptual framework for AI systems engineering.

- **Verified safety compliance layer:**
  A system is assumed to be acceptable safe only when the involved stakeholders gain enough confidence in the emergent safe behavior. Traditionally, a system is usually verified by formal models and various types of testing. QA experts and users play a key role to come to a consensus regarding the level of safety of the system. However, safety assurance and verification have become extremely challenging with the inherent uncertainty and data dependency of ML models. Therefore, at this layer, ML experts, HCI experts, data scientists play a crucial role not only to validate training and testing data but also to evaluate overall robustness, scenario coverage, performance, etc. of the ML-based component.

All the above-mentioned layers are based on a layer of basic factors, like assumptions, level of autonomy, regulatory constraints, financial constraints, ethical constraints, data constraints, etc. This framework provides better visualization of how a safe output space of AI systems is designed across the layers starting from the initial naive problem space.

5.1. Layer-1: Challenges and research efforts

In the last few years, researchers started discussing the challenges faced by software engineers while working on ML techniques for complex AI systems (Ishikawa and Yoshioka, 2019; Arpteg et al., 2018; Amershi et al., 2019; Sculley et al., 2015). As most of the current intelligent systems are software-intensive, we choose to analyze the challenges from a software engineering perspective in detail. Software research community has always asserted the importance of initiating the engineering process by having comprehensive understanding of the problem space and expectations of customers (users). We reflect the same understanding in our layered framework (Fig. 3). In order to successfully explore the safe solution space in Layer-3, the first step should be to start with exploration of the rich and relevant problem space. Therefore, in this subsection, we explain the challenges faced by requirements engineers, domain experts, and data scientists in the early phase of engineering.

(1) Shortcomings of traditional requirements engineering-driven approach:

With the advent of deep learning, the traditional end-to-end RE process became futile. It is hard to make users' expectations explicit when the operating environment itself is extremely uncertain. Stating all possible 'shall' statements in the early phase of engineering is impractical with the inherent uncertainty and lack of clarity on the relevant context. Bosch et al. discuss that requirements-driven approaches may need to be complemented by other approaches (Bosch et al., 2018). The authors identify three distinct approaches that may coexist during the development of intelligent software:

(i) RE-driven: Development according to well-understood specifications. This approach may be used only when the new feature is well understood.

(ii) Output/data-driven: Development according to a given quantitative target. The goal here is to improve the metric by experimenting with different solutions.

(iii) AI-driven: Development of components using ML/DL based on an already available large dataset.

The authors envisioned a holistic integrated development approach (DevOp) incorporating all the above-mentioned approaches in which the system consists of traditional software and AI components and also there is scope for continuous deployment. However, this integrated framework requires further validation and comprehensive guidelines for its usage. We need to further understand the real challenges at Layer-1 by thoroughly investigating the core issues from RE perspective.

(2) Understanding the problem domain:

Problem domain understanding is equally important in the case of traditional and intelligent system engineering practices. It is always one of the most complex activities during the early phase of engineering. The complexity is even higher in systems with ML components where the system learns about the domain-specific concepts through training data (Ishikawa and Yoshioka, 2019; Belani et al., 2019). Rahimi et al. (2019) explain this challenge using an example from an automated pedestrian collision avoidance system. The concept of 'pedestrian' is not clear in terms of what it means in a particular context. A person walking a bike (instead of riding it) or a person in a wheelchair should also be included in the semantics of pedestrians to ensure safety. The author suggested that a thorough understanding of the domain is possible by:

(i) Benchmarking the domain

(ii) Interpreting the domain dataset
(iii) Interpreting the domain learned by ML model for further validation

The authors proposed to create a domain ontology based on the web search result of the domain concept ('pedestrian' in the example). This ontology can be further used to assess whether the training data represents the problem domain as depicted in the ontology by extracting features from the dataset. Incorrect or missing correlation between ontology element and the dataset can be helpful in identification of the gap or incompleteness of the dataset.

(3) Setting quantitative targets (functional requirements):

The main objective of RE activities for any system is to state the users' expectations explicitly. However, setting explicit quantitative targets (which are often referred to as functional requirements of ML component) is a great challenge, not only because it is hard to declare desired output, but also it is difficult for users to understand ML related performance metrics such as recall, precision, etc. (Vogelsang and Borg, 2019). Data scientists need to take part in RE actively to help the client to set a metric that is not too technical, yet scientific enough to measure the performance. For example: 'Lift' (improvements of performance) can be explained by introducing a comparison of ML predictions to random predictions. More about these topics are discussed in the upcoming sections on setting targets for validations and testing.

(4) Setting qualitative targets (non-functional requirements):

While establishing quantitative targets is already gaining the attention of the research community, a few researchers are also discussing qualities or nonfunctional requirements associated with the functional requirements (Köhl et al., 2019; Horkoff, 2019). It is argued by Horkoff, in today's time we are unknowingly relying more on quality than quantity to gain trust (Horkoff, 2019). For example: rather than relying on 99.99% of the accuracy of the target to ensure safety, more efforts need to be invested in justifying transparency, testability, explainability, reliability, fairness, etc. She also mentioned that further research needs to be conducted regarding a thorough understanding of the new definition, catalog, target, and trade-offs for NFRs for different kinds of ML algorithms used in intelligent systems. Nakamichi et al. (2020) made a recent contribution by proposing a quality model, quality characteristics, and a measurement method for ML-based software systems. One of the most important qualities from the safety perspective is the robustness of the ML model, especially for perception tasks where a slight change in the input can cause misclassification. Hu et al. (2020) proposed how to formally specify robustness requirements for such ML-based component by identifying the invariant and equivariant. Each robustness requirements mainly consists of three components: a formal definition of transformation over the inputs, a range of values for transformation parameters, and a corresponding indication of invariant and equivariant. However, deriving an exhaustive list of invariant and equivariant is an open area of research for ML-based components deployed in an uncertain environment.

(5) Lack of requirements analysis and modeling techniques to address uncertainty:

To analyze the intrinsic uncertainty and unpredictability of ML models, concrete analysis and modeling technique is required to be performed during RE phase. However, not enough attention has been paid to end-to-end modeling and analysis methods. Recently, F. Ishikawa proposed a goal-based evidence-driven RE modeling to analyze the goals and unresolved uncertainty of ML models (Ishikawa and Matsuno, 2020). The authors explained how the model can depict the evidences captured by experiments and operation to validate or invalidate the feasibility or performance level. More efforts need to be invested in developing automated tools to link between decision making by the ML models and the collected evidences during testing and operation.

(6) Explaining 'black-box' to users:

'Safety' and 'trust' are correlated. Lack of transparency/explainability leads to weak assurance of safety and as a result users or the society as a whole lack trust in such systems. Therefore, some of the literature emphasizes the importance of the explainability of AI systems. Köhl et al. (2019) report that a lack of understanding of some phenomenon motivates certain groups of stakeholders to seek explanation in a certain context. Therefore, we can consider including explainability requirements as a new addition to the non-functional requirements family. While eliciting explainability requirements, other requirements, such as security and cost, may conflict with them. This will eventually lead to exploring the rich problem space even further by performing trade-off analysis. Vogelsang and Borg mentioned that in case of ML, explainability is twofold: explain what has been learned and how each prediction has been made by the model (in the context of the predictive model) (Vogelsang and Borg, 2019). Although explainable AI systems have been garnering attention recently, the ML research community is yet to formalize situations that should demand an explanation. Moreover, further research is necessary on the appropriate level of abstraction to attain the required explainability of a system.

(7) Declaring hidden consumers:

For any system, it is of the highest priority to know the target users and consumers of the output. Complex AI systems with one or more ML components are often developed without a proper understanding of who is accessing the output (e.g., the prediction output directly or log files indirectly later) (Arpteg et al., 2018). Inadequate access control strategy can have a significant impact on the overall emergent safe behavior of the systems as undeclared consumers may use the output in an unintended way (Sculley et al., 2015). Belani et al. express their concern over undeclared consumers as this hidden risk factor has not been sufficiently scrutinized (Belani et al., 2019). We believe that identifying and documenting the right set of consumers are essential activities at Layer-1 to safeguard the system from unintended usage in the future.

Discussion:

We believe that research on RE for AI systems is at its beginning. Only a limited part of the literature has discussed the challenges faced at an early stage of AI systems engineering. Instead, research has focused on the technical excellence of AI algorithms and their applications. However, as we discuss the challenges at Layer-1 in this subsection, we summarize the current understanding of RE and SE research community for AI systems in Table 4. This table shows the latest guidelines for each of the phases of RE and specifies recent advances on new types of requirements that should be part of the requirements specification in the days of AI.

5.2. Layer-2: Challenges and research efforts

In this section, we focus on the analysis and design issues that are closely related to safety concerns. As shown in the framework, in Layer-2, safety engineers, designers, developers, ML experts are required to collaborate to explore the safety constrained design space. Therefore, we will discuss risk analysis, safety requirements, safety standards of autonomous systems.

(1) Safety-driven design of complex systems:

The preferred way to build a safe system is to consider safety from the very beginning of requirement engineering. Firesmith described a taxonomy for four kinds of safety requirements named as — pure safety requirements, safety-significant requirements,

Table 4
Summary of state-of-the-art RE approaches for safe and trustworthy AI.

RE activity	Guidelines
Requirements Elicitation (Ishikawa and Yoshioka, 2019; Rahimi et al., 2019; Belani et al., 2019; Nakamichi et al., 2020)	-Include data scientists and legal experts – Benchmark the domain – Elicit new data sources – Identify sensitive/protected features of the data – Situations that demand an explanation to help users
Requirements Analysis (Horkoff, 2019; Ishikawa and Matsuno, 2020)	– Discuss performance measures that are easily understandable for users – Conditions for data pre-processing, cleaning, etc. – Required level of automation needed in the process to meet the constraint – Evidence-driven goal-based requirements modeling and analysis (Insufficient attention is paid to requirements analysis activities)
Requirements Specification (Rahimi et al., 2019; Köhl et al., 2019; Horkoff, 2019; Vogelsang and Borg, 2019; Nakamichi et al., 2020; Hu et al., 2020)	– Data requirements – ML model requirements – ML process requirements – Quantitative targets – Measurable qualitative targets – Explainability requirements – Ethical and legal requirements – Robustness requirements
Requirements V & V (Rahimi et al., 2019; Hu et al., 2020)	-Confidence in data – Data dependencies – ML process requirements – Quantitative targets – Robustness targets (Not enough work is done on requirements validation)
Requirements Evolution (Ishikawa and Yoshioka, 2019; Amershi et al., 2019)	– Documenting dataset and model versions – Data dependencies – Requirement-data-feature-output traceability (Insufficient attention is paid to requirements evolution)

safety constraints, and requirements for safety systems (Firesmith, 2004). However, elicitation of safety requirements was out of the scope of this work. To the best of our knowledge, the necessity of a change in the traditional accident model due to the emergence of software was first identified by Leveson (2004a). The authors explained how the traditional accident model could be enhanced by accommodating social, cultural, and organizational aspects. A complete hazard analysis methodology named STAMP (System Theoretic Accident Model and Process) was proposed by researchers in Stringfellow et al. (2010b) and Leveson (2004b), in which accidents were viewed as control flaws (failure in the interactions among the components) instead of a component failure. This methodology not only helped to analyze the source of hazards and deriving safety constraints but also helped to make appropriate design decisions to enforce the derived constraints. This safety-driven design approach STPA (System Theoretic Process Analysis) is explained in Owens et al. (2008) with the help of a case study on spacecraft. Although this work encompassed the socio-technical factors influencing the design of a safe system, it precludes the hazards caused by human error. Moreover, this work does not exclusively address the inherent uncertainty and opacity of ML-based software-intensive systems. Therefore, we will not probe this research further.

Kuper et al. introduced the concept of verification-friendly design of neural networks (Kuper et al., 2018). Varshney explained how four common safety strategies (inherently safe design, safety reserves, safe fail, and procedural safeguards) can be mapped to an ML context (Varshney, 2016). The authors emphasized that ML models are very different in terms of the inherent uncertainty in train and test data, and their probability distribution. Rejecting a less confident decision made by the mode can be an option to fail safely. However, it is important to define the decision boundary carefully as the distance from the boundary is not always inversely proportional to confidence. A part of the input space with low density can contain much epistemic uncertainty as the boundary may be based on inductive bias. In the same direction, Gu and Easwaran proposed Feature Space Partitioning Tree (FSPT) to partition feature space and to reject input instances from the low-density feature space (Gu and Easwaran, 2019). In the case of reinforcement learning, safe outcome has been confirmed by introducing risk-aware policies and rewards (Chow et al., 2017), safety-aware planner (Rong and Luan, 2020), safety supervisor (Chen et al., 2020), etc. Although these contributions were very significant in terms of ensuring safety at the ML component level, they lack insight into how the safety standard can be defined and mapped against system or sub-system level safety goals.

(2) Lack of method to model uncertain environment including human behavior:

For safety-critical AI systems, combinations of numerous variables lead to an unlimited number of situations to model, which results in an impractical verification process. The complexity amplifies when human is in the loop. Human's behavior is mostly variable and uncertain. Therefore, for semi-autonomous systems, the problem of environment modeling is two-fold. Research on verification of AI systems can get momentum only if there is a well-understood technique to model the uncertain environment. As complete formal modeling of the uncertain environment seems to be challenging by many researchers, attention is paid towards introspective modeling of the environment. For example: identifying and analyzing the assumptions that a system makes about its environment can be helpful to verify whether the system is capable of monitoring the right variables or not. Seshia et al. have addressed such issues with the help of control

Table 5
Overview of the safety standards and the relevant autonomous systems.

Safety standards	Autonomous systems
ISO/DIS 3691-4 (ISO, 2020)	Driverless trucks
ISO 13482 (ISO, 2014)	Personal Care Robots
ISO 19014 (ISO, 2018b)	Earth moving machinery
ISO17757 (ISO, 2017)	Earth moving machinery and mining
ISO 18497 (ISO, 2018a)	Agricultural machines
IEC 62267 (IEC, 2009)	Automated urban guided transport
ISO 26262 (ISO, 2018)	Road Vehicles
ISO/PAS 21448 (ISO, 2019)	Road Vehicles

theory. Extraction of monitorable assumptions is proved to be feasible for simple controllers (Li et al., 2011; Ghosh et al., 2016; Li et al., 2014). To address the issues of modeling unpredictable human behavior, one way could be gathering data about real and simulated environments to learn about the environment model. Some researchers have shown it to be effective for verification and control of an autonomous vehicle to generate human behavior models from driving simulators and human subject experiments (Sadigh et al., 2016b, 2014, 2016a).

(3) Scenario-based safe design and development of autonomous systems:

For safety-critical systems like driverless cars, deployed in an uncertain environment, collecting sufficient safety requirements is an exhaustive task. The risks and potential hazards are mainly hidden in the operational world and its numerous variations. Therefore, very recently, researchers started focusing on purposefully varying the operation scenarios of systems to elicit safety requirements in the early phase of the system life-cycle. Bach et al. (2016) presented a methodology for a model-based design of scenarios from real-world test data. Abstraction of temporal and spatial information act as the key enables of not only coherent modeling of scenario but also support specification of requirements and derivation of test cases. In a similar direction, Menzel et al. (2018), proposed three levels of abstraction for scenarios along the V-model of development process following ISO-26262 (this standard will be discussed later). These researchers elaborated the concept of 'Scenario' and defined three layers of abstractions of scenarios:

(i) Abstract functional scenario in concept phase: Involves identifying semi-formal hazardous scenarios including operational scenarios and malfunctioning behavior.

(ii) Detailed logical (technical) safety scenarios in development phase: Involves describing scenarios including parameter ranges of the state values that are used to represent functional scenarios.

(iii) Concrete scenarios for validation and verification: These scenarios represent the operational scenarios with concrete values of each parameter. Proposed process steps of the usage of such safety concerned scenarios can be very helpful to generate the artifacts at each step and to maintain traceability among them.

(4) Safety standard to guide for analysis and design process:

Standards are believed to be one of the best ways to guide and assess (in later phases) the development of a particular system through specification. Especially for AI systems, standards can provide explicit specification/requirements for explainability, robustness, fail-safe design (Cihon, 2019). ISO and IEEE are the two leading bodies that have been developing various standards regarding safety requirements of autonomous machinery. Table 5 summarizes some of the relevant standards in this area. However, none of them exclusively considers AI. The use of ML commences an overall paradigm shift in the design and development process. Unfortunately, these standards fail to ML-specific concerns. Nevertheless, some of the safety considerations to ensure safety are rightly identified by Google (Google White Paper, 2019).

1. Is the objective function appropriate?
2. Has the exploration space been sufficiently constrained?
3. Does the model's training reflect the current real world?
4. Can the risk of data poisoning be mitigated?
5. Has the AI system been adversarially tested?

Best practices to address these concerns should be collected in formal standards. The formulation of safety standards entails risk analysis, risk control, and risk monitoring (Ozlati and Yampolskiy, 2017). These standards are mostly under development. Formalizing AI safety standards is an ongoing endeavor. Many researchers are working on the same from different perspectives. For example, Ozlati and Yampolskiy (2017) considered the diversity of AI systems and discussed that different categories of AI systems may have diverse risks and mitigation strategies. Therefore the authors suggested using a modified Delphi methodology study as a starting point of a standing body that can develop and evaluate AI safety standards under AI SDO. Keeping the diversity of AI systems in mind, the authors recommended that the modified Delphi study should cover separate risk assessments for different system categories. Luo et al. focused on environment-centric safety requirements of automated unmanned systems (Luo et al., 2019). Environment safety requirements are elicited from the entities of the environment — other systems, human, constraints. The authors classify such requirements along MAPE-K process. After conducting a literature survey, the authors conclude that few gray areas need further research, such as methodology to solve safety concerns (collision avoidance), optimizing safety constrained learning technique (MDP), etc. Most discussed domain-specific standards in the days of AI: Recently, the automotive industry has made significant progress in developing and testing driverless cars. However, as of today, neither the industry nor the government can fully assess the safety of self-driving cars. Therefore, there is a sudden rush to set standards for such AI-based autonomous vehicles among the public organization like IEEE and also public sectors like Safety First for Automated Driving (SaFAD) led by Audi, BMW, etc. In this subsection, we will explain some of the standards related to an autonomous vehicle in detail.

ISO 26262. ISO 26262 is the derivative of IEC 61508 (ISO/IEC, 2010a; ISO, 2018). It mainly covers automotive development, production, and maintenance of safety-critical systems. The key component of ISO 26262 is the automotive safety lifecycle (ASL) which describes the fundamental concepts of safety plan, safety manager, safety review and audit. ASL consists of six phases: management, development. production, operation, service and decommission. This standard also defines the automotive safety integrity level (ASIL). Based on safety analysis of the critical functions of the system, a risk analysis is performed. The risk analysis combines the probability of exposure, controllability of a driver, the severity of outcome resulting in ASIL from A to D. As per this standard, development and verification practices should correspond to the corresponding ASIL. However, as argued by Borg et al. (2018), with the advent of ML algorithms (especially less transparent DNN), traditional standards fall short. Salay et al. analyzed ISO 26262 from ML perspective to identify the key factors of conflict (Salay et al., 2017). The researchers found gaps in the software development requirements of ISO 26262, and proposed requirements to fill those gaps. One of the major contributions of their work is the elaboration on connecting data to safety concerns while working with ML algorithms. Prior knowledge of the function to be performed by ML component plays a significant role in fostering safety. The ways input can change without affecting the output are termed as invariants. Invariance to lighting level, positions, etc. is directly associated with the safe outcome of the learning model. Similarly, equivariants describe types of change in input that should result in a particular type

of change in the output. Along with these two specifications, different kinds of constraints like probabilistic, pattern-based, and context-based constraints on the input and output and control the safe behavior of ML component. Moreover, analyzing data distribution, its coverage of edge-cases also enable developers and safety engineers to assess the expected safe behavior of the AI-based system. The researchers also mention that model selection, feature selection, training, and testing specification are the key artifacts in arguing the safety case of an ML algorithm.

UL-4600. Koopman et al. working with UL (Underwriters' Laboratories) proposed an initial draft of UL-4600 standard for fully autonomous vehicles (Koopman et al., 2019). To the best of our knowledge, UL-4600 is the latest and the most advanced standard that aims to address the challenges of the full autonomy of HAV. The traditional standards lack flexibility as they are usually census-based and updated every 5 to 10 years. However, in the days of AI, developers are exploring new technologies rapidly to provide a better solution to the concerned problem. Therefore, flexible standards like UL-4600 gained attention in recent times. UL-4600 does not prescribe direct guidance to the proper development process. Rather, it guides on building the safety case (which will be discussed later in the next section) for HAV (Yoshida, 2019). A safety case is a very important artifact for safe systems design. Identified topics that are planned to be addressed in this standard are:

(i) Definition of operational design domain (e.g., weather, scenarios, etc.)
(ii) ML faults (e.g., training data gaps, etc.)
(iii) External operation faults (e.g., fault of other vehicles, etc.)
(iv) Faulty behavior of the non-driver humans (e.g., pedestrians, etc.)
(v) Non-deterministic system behavior (e.g., test planning, etc.)
(vi) High residual unknowns (e.g., requirements gaps, etc.)
(vii) Lack of human oversight (e.g., passenger handling, etc.)
(viii) System-level safety metrics.

This standard emphasizes that it is more effective to continually evaluate and improve the residual risk present in the system than to conform to a standard during deployment. Therefore, developers need to be actively involved and take responsibility for safety risk identification and self-assessment over the iterations of the development life-cycle.

Discussion: In 2020, several new standards from various sources for AI-based systems are likely to be rolled out; for instance, ISO/IEC PAS 21448:2019, ISO/IEC CD 23053, ISO/IEC AWI 23894, ISO/IEC AWI TR 24027, ISO/IEC AWI TR 24028, ISO/IEC DTR 24029-1, ISO/IEC CD 38507 (ISO/IEC, 0000c; ISO, 0000a; ISO/IEC, 0000d,b; ISO, 0000b; ISO/IEC, 0000f,e). We could not discuss these standards in detail as most of the standards are still under development and not many open-access documents are available regarding these standards. However, as Riccardo Mariani (Yoshuda, 2020) mentioned that, with the arrival of the new safety standards from various sources, it is expected that those standards will be written from diverse perspectives of AI-based systems. Therefore, the main goal should be to analyze those standards and minimize the overlap.

5.3. Layer-3: Challenges and research efforts

As rightly argued by J. Morton et al. Trust is the prime concern that needs to be established before a complete release (Morton et al., 2018). Many researchers focused on validation and verification of the system life cycle to prove that the system is acceptably safe enough to be deployed. In this subsection, we will explain the challenges to perform end-to-end validation and verification of AI systems. Seshia et al. explained the shortcomings of the current practices and proposed to move towards the paradigm of Verified Artificial Intelligence (Seshia et al., 2016). Further, we discuss the directions that researchers have been following recently to address those challenges.

(1) Lack of formal specification to verify the system:

Traditional formal verification is mostly founded upon strong mathematical statements of the way system should behave. However, for ML-based systems, it is extremely difficult to describe the expected behavior precisely in a mathematical way. For instance: there is no concrete method for a module of an autonomous car that uses computer vision to perform object recognition, human–object classification. As it seems to be nearly impossible to formally specify the exact behavior of any ML component, few researchers have tried to solve the problem from a different perspective. Instead of formalizing the component level behavior, end-to-end system-level behavior specification can be used for verification purposes. Seshia et al. suggested that specification mining techniques can be used in such cases (Seshia et al., 2018). Jan Leike et al. presented a suite of reinforcement learning environment to assess the conformity with the intended safe behavior (Leike et al., 2017). The authors also classified various safety problems in the case of RL. For example, specification problems, safe interruptability, reward gaming, safe exploration, etc. However, the cost of verification and validation of such components is not trivial. Especially for domains like automotive systems, aircraft systems, verification approach is not scalable for the real world unless they are designed that way. However, more research is required to ensure that the intention of the designer is rightly articulated into the ML component through well-designed cost and reward functions.

(2) Lack of system modeling approaches for data-driven ML components:

Formally modeling complex deep neural networks with millions of data, several layers, stochastic behavior, and hundreds of features to learn poses a challenge. To model such ML-based components, an explanation based on generalization and abstraction is needed and both input and probabilistic (uncertain) output need to be formally modeled and explained. Furthermore, such uncertain output and its corresponding effect on the system-level specification needs to be formally modeled. More significant research on formalism of Markov Decision Process (Nilim and El Ghaoui, 2005), probabilistic logics (Sadigh and Kapoor, 2016; Sadigh et al., 2014), and counterfactual reasoning can lead to the mitigation of the problem of systematic modeling of data-driven ML component.

(3) Insufficient method for quantitative verification:

Apart from formal specification and modeling of ML components, training and testing data play a very important role in verification of the ML algorithm or component. The behavior of a frozen learning model (with no continuous learning strategy) may vary with a small perturbation of the test data, These adversarial perturbations pose a new challenge to verifying ML-based component and using them as a part of safety-critical systems such as autonomous cars (Fawzi et al., 2018; Nguyen et al., 2015; Moosavi-Dezfooli et al., 2016; Goodfellow et al., 2014). Moreover, the traditional boolean outcome of the verification and validation process is inadequate for ML. Quantitative requirements (data requirements, equivariant, and invariant specification, etc.) can contribute to design a quantitative verification process. Semi-autonomous systems with both machine and human controllers can be considered as hybrid systems (Ghosh et al., 2016; Dennis et al., 2016; Alur, 2011) and therefore can follow a probabilistic process of verification. As suggested by Seshia et al. in Seshia et al. (2016), randomized formal methods to systematically generating training and testing data can be one of the options to move towards formal verification of ML algorithms.

Randomized formal methods need to be improved to address the constraints on the legal input and output space. Randomness requirements can define the output distribution. More rigorous research on constrained random sampling is expected to aid the process (Chakraborty et al., 2014). Additionally, SMT solving can also be extended by combining with optimization problems to handle similar issues (Shoukry et al., 2017; Tuncali et al., 2018b). An in-depth survey verification of neural networks can be found in Xiang et al. (2018).

Recent surge in research on safe deep neural network (DNN)

Safety assurance of neural networks in particular has received much attention in the last few years. Various strategies of testing have been adopted by researchers to gain the confidence that the neural network can be safely used in safety-critical systems; for instance, white box testing (Lee et al., 2020; Pei et al., 2017), gray box testing (Tian et al., 2018), feature-guided black box testing (Wicker et al., 2018), mutation testing (Wang et al., 2019; Ma et al., 2018b), concolic testing (Sun et al., 2018b), etc. Researchers have also paid attention to the testing criteria suitable to ensure safe output of a DNN-based system (Ma et al., 2018a; Sun et al., 2018a). Byun and Rayadurgam proposed a manifold-based ML testing framework (Byun and Rayadurgam, 2020). The authors argue that compared to neuron coverage manifold-based coverage is a more effective measure of assurance. In the same direction, another recent study by Harel-Canada et al. argued the unsuitability of neuron coverage as a V & V metric for neural networks (Harel-Canada et al., 2020). The assessment showed that the increase of neuron coverage may rather hinder the way of generating an effective test suite for neural networks. Falsification approach (testing against corner cases) has also gained interest among researchers as this approach can successfully analyze the situation where a system may fail (Dreossi et al., 2019; Tuncali et al., 2018a). In case of formal verification, numerous approaches have been adopted by the researchers. For example, input–output range specification-based verification (Dutta et al., 2017; Ehlers, 2017; Huang et al., 2017a; Katz et al., 2017), solver-based verification (Sun et al., 2019), reachability-based verification (Fazlyab et al., 2019; Xiang and Johnson, 2018; Ivanov et al., 2019; Akintunde et al., 2018), etc.

(4) Difficult to perform rigorous run-time testing to address the uncertainty:

Morton et al. proposed a method for deriving close-loop testing strategies for safety-critical systems (Morton et al., 2018). Koopman et al. made remarkable contributions in the field of safety validation of automated vehicles (Koopman and Wagner, 2018; Koopman, 2018b). The authors proposed a phase-wise testing approach that not only mitigates the risks but also identifies the assumption violations and the unexpected situation at runtime (Koopman and Wagner, 2018). Researchers have also focused recently on safety verification of autonomous systems with neural networks-based controller at the system-level in the presence of hardware faults like lidar faults etc. Ivanov et al. (2020), Hoang et al. (2020). Although this approaches can be helpful to reduce the testing load, more thoughts need to be given towards the traceability of the testing artifacts from the system-level to the component-level.

(5) Difficult to evaluate robustness against adversarial attacks on ML-based systems (safety meets security):

In the last few years, many researchers have expressed their concerns about the potential threats to the stability of ML-based systems posed by adversarial attacks. It is difficult to predict such attacks, and it is also complicated to model the response of a component or system as a whole to such attacks. Therefore, to develop a reliable system it is imperative to not only analyze the inherent risks of ML-techniques, but also to protect them from intentional adversarial attacks. To ensure functional integrity of modern ML-based systems safety and security related knowledge should be used in combination. Szegedy et al. Biggio et al. first paid attention to adversarial examples that can easily deceive neural networks (Szegedy et al., 2013; Biggio et al., 2013). If the attacker adds a small perturbation to an image with the proper calculation, a well-trained neural network (NN) can misclassify the image with surprisingly high confidence. Such unstable behavior can lead to catastrophic consequences in case of safety-critical systems that depend heavily on computer vision to take decisions. Numerous recent works have analyzed different ways to fool deep neural networks (DNN) (Nguyen et al., 2015; Papernot et al., 2017; Grosse et al., 2016; Papernot et al., 2016). Bastani et al. proposed metrics to measure robustness against such adversarial examples (Bastani et al., 2016). Very recently, Naseer et al. proposed formal methods to analyze noise tolerance, training bias, and input sensitivity of neural networks (Naseer et al., 2020). Similarly, RL agents can also be heavily manipulated by malicious attacks. Huang et al. (2017b) have recently shown that RL-algorithms such as DWN, TRPO, A3C can be vulnerable to malicious inputs. Even a small perturbation can lure an RL agent to move to an undesirable state and take unsafe action. White and black-box attacks have been well investigated in this work. In case of white-box attacks, the attacker is assumed to have access to the policy network. Whereas, black-box attacker have only partial or no such information. The researchers show that white-box attackers are more effective than black-box attackers. It is possible to confuse an RL agent with trained policy even in real-world black-box scenario. For instance, lane following policy of an autonomous car can be deliberately altered by placing a small mark on the road surface or road signs.

Defenses against adversarial attacks:

Various defense mechanisms have been proposed to ensure robustness of ML models. We briefly consider the two most discussed defense mechanisms.

(i) Adversarial training:

This is a brute force process of generating as many adversarial examples as possible and using them to train the model in advance. Since Szegedy et al. (2013) showed the existence of adversarial examples to fool DNNs, many researchers focused on ensuring robustness of learning models from such adversarial attacks (Papernot et al., 2017; Gu and Rigazio, 2014; Shaham et al., 2015). As discussed by Hazan et al., dealing with adversarial perturbation can eventually help to optimize a ML model in a more robust way (Hazan et al., 2017). One of the major challenges faced in training DNNs with adversarial examples is synthesizing a sizable number of adversarial examples. Various ways have been proposed to synthesize such examples in whitebox settings (Moosavi-Dezfooli et al., 2016; Goodfellow et al., 2014; Szegedy et al., 2013; Papernot et al., 2017; Carlini and Wagner, 2017; Chen et al., 2017). Another challenge is to generate physical-world adversarial examples (2D photos, adversarial patches, 3D prints, etc.). These physical or real-world adversarial examples have been demonstrated for various domains such as face recognition, image classification, speech-to-text, etc. (Kurakin et al., 2016; Carlini et al., 2016; Eykholt et al., 2018).

(ii) Defensive distillation:

In this defense mechanism, a model is trained to provide outputs in terms of probability instead of hard labels. The distillation process reduces the gradients used to create the adversarial example. As a result, this defensive mechanism of a DNN can reduce the effectiveness of adversarial sample from 95% to less than 0.5%. Detailed analysis of the state-of-the-art defensive measures can be found in recent surveys on robust deep learning (Ozdag, 2018; Chakraborty et al., 2018; Xu et al., 2020).

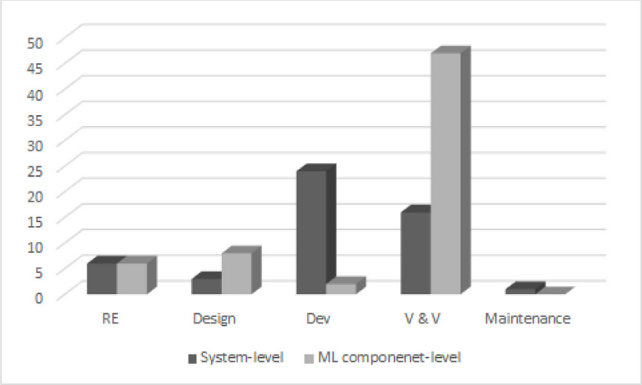

Fig. 4. Number of primary studies along the phases of SE activities at system-level and ML component level.

(7) Inadequate practice of demonstrating assurance cases for ML-based system:

Despite ML showing promising improvements in performance due to its introduction in many complex systems, it will always lack confidence of the users and society unless a proper assurance case is provided. However, unfortunately, there is currently no formal practice of producing end-to-end safety assurance cases. Palin et al. (2011) proposed patterns for designing safety cases covering all aspects of ISO 26262 for the automotive domain. However, these proposed reusable safety arguments did not consider the complexities that ML techniques bring. A fault-free system can still behave in an unintended way due to the intrinsic uncertainty of ML techniques. Therefore Gauerhof et al. proposed a safety assurance case for a pedestrian detection function using Graphical Structuring Notation (GSN) (Gauerhof et al., 2018). In this approach, the risks of under-specification (unclear tasks and environments), deductive gap (incorrect learning of features from insufficient data), and semantic gap (unclear domain concepts) have been reduced by defining corresponding arguments and evidences. This type of well-structured validation targets helps gain confidence. Matsuno et al. (2019) explained how the inherent uncertainty of ML models affects the strategy and activities for safety assurance. According to these researchers, continuous argument engineering is useful in such cases to determine the weakness of a model, which can never be pre-calculated. They also developed tool support to assess and track the latest state of assurance. A similar idea was previously proposed by Denney et al. (2015) and it was applied to the aviation system domain. The authors showed how safety management system needs to manage the safety cases not only during the development and deployment period but also at run-time based on real-time operational data. However, this approach did not explicitly handle the risks and uncertainty of ML models. Recently, patterns for arguments safety assurance of ML in the medical diagnosis system have been proposed as well (Picardi et al., 2019; Picardi and Habli, 2019). In these approaches, the researchers provided a detailed structure of assurance cases using GSN while considering medical settings and ML activities. These patterns can be widely used in other safety-critical domains, given sufficient settings or context-specific information about the domain.

Discussion. Despite putting rigorous efforts on V & V of ML-based systems, as argued by Brundage et al. not having a formal consensus on metric to measure general property like safety of these systems, there will be always a tension between the verifiability claims and the generality of such claims (Brundage et al., 2020).

6. Discussion of research questions

In this section, we revisit our research questions to finally map the primary studies to answer the research questions from an engineering process perspective. Although in the previous section, we discussed the challenges and research efforts along each layer, we are going to summarize our findings from the literature review in this section. For the ease of understanding and to have a better traceability, we first make a list of the primary studies and assign a unique ID to each of them as shown in Table 6. In this table, we also summarize the perspective, domain, objective and specific ML techniques that each of the studies focuses on. In case of a paper written for (or tested on) a specific domain, we specify the name of the domain. Otherwise, we mention it as "General". Similarly, we mention NS (non-specific), if the study is not exclusively designed for any particular ML technique. DNN, RL, NN stand for Deep Neural Networks, Reinforcement Learning, and Neural Networks respectively.

6.1. RQ1: How can we easily comprehend the complexity and challenges involved in fostering safety of complex intelligent systems?

To visualize the complexity and challenges faced by multiple stakeholders involved in the process of engineering safe AI systems, we analyzed the primary studies based on the proposed three-layered framework depicted in Fig. 3. In Table 7, we map the primary studies to each layer of the three-layered framework. Compared to the other two layers, the Problem Definition Layer has received less attention. As this layer involves activities like analyzing the problem to be solved, understanding the domain, setting the right targets, it is imperative to focus more on this layer to avoid late realization of setting incorrect targets and objectives. As the table shows, many researchers are focusing on the validation and verification-related challenges in the third layer. While we acknowledge the absolute necessity of formal verification of ML-based systems to gain trust, we also believe having a clear idea about "what to verify", "against which metrics to verify", "what are the qualitative and quantitative targets" is equally important to verify and validate a system in the right way. In layer-2, more work needs to be done on modeling uncertain environment from a software engineering perspective. Most of the standards included in the review need to be updated to accommodate the new challenges that are brought in by the advent of ML algorithms.

In a nutshell, with the help of the three-layered framework, we can easily observe that a lot of areas are still open for research to strengthen a strong foundation (layer-1 and 2) of the overall system engineering of complex ML-based systems.

6.2. RQ2: How have safety concerns been addressed by the researchers along the phases of SE process?

Summary of the state-of-the-art safety approaches at the system-level and ML-based component level along the phases of software engineering life-cycle is shown in Table 8 and Fig. 4. We did not map (Ishikawa and Yoshioka, 2019; Arpteg et al., 2018; Amershi et al., 2019; Salay et al., 2017; Lwakatare et al., 2019; Varshney, 2020) as these papers do not exclusively focus on any of the particular SE activities. Instead, these studies focus on an overall engineering process of developing ML-based systems. The reason why no primary study could be directly mapped to the development phase for ML-based components is that in case of ML, the line between the design and development is very thin. The key concept here is to take safe design decisions while training the ML model. For example, safe exploration, safe policies, safety-aware rewards, etc. All these approaches are

Table 6
List of the primary studies.

ID	References	Perspective	Domain	ML-technique	Objective
P001	Ishikawa and Yoshioka (2019)	Software engineer	General	NS	Understanding engineering process-related challenges
P002	Arpteg et al. (2018)	Software engineer	General	Deep learning	Understanding engineering process-related challenges
P003	Amershi et al. (2019)	Software engineer	General	NS	Understanding engineering process-related challenges
P004	Rahimi et al. (2019)	Software Engineer, Domain expert, Requirements engineer	General	NS	Specifying concepts in safety-critical domain to reduce conceptual uncertainty in ML-based components
P005	Köhl et al. (2019)	Software Engineer, Domain expert, Requirements engineer	General	NS	Unifying the notion of explainability in case of ML-based systems
P006	Sculley et al. (2015)	Software engineer, system engineer	General	NS	Exploring ML-specific risk factors for maintainable ML-based systems from engineering perspective
P007	Bosch et al. (2018)	Software engineer, AI expert,Requirements engineer, Data scientist, system eng	General	NS	Discussing holistic approach of software development in the days of AI
P008	Horkoff (2019)	Software Engineer, Requirements Engineer	General	NS	Setting qualitative targets for ML-based systems
P009	Vogelsang and Borg (2019)	Data scientist, Requirements Engineer	General	NS	Understanding RE challenges for ML
P010	Belani et al. (2019)	AI expert, requirements engineer, system engineer	General	NS	Defining RE taxonomy for AI
P011	Ishikawa and Matsuno (2020)	AI expert, requirements engineer, system engineer	General	NS	Discussing goal-based requirements engineering methods to address intrinsic uncertainty of ML
P012	Nakamichi et al. (2020)	AI expert, requirements engineer, system engineer	General	NS	Defining quality model, quality characteristics and measurements for ML-based systems
P013	Hu et al. (2020)	Software Engineer, Domain expert, Requirements engineer	General	NS	Specifying and testing robustness requirements of ML-based components
P014	Cihon (2019)	Governance	General	NS	To reframe international standards as tools of AI policy.
P015	Google White Paper (2019)	Governance	General	NS	Discussing concrete issues on AI governance
P016	Ozlati and Yampolskiy (2017)	System engineer, AI expert	General	NS	Adopting formal risk assessment practices for safe AI development
P017	Luo et al. (2019)	System engineer, safety engineer	Unmanned automated systems	NS	Setting up a taxonomy of environment-centric safety requirements for automated unmanned system
P018	ISO (2020)	NA	Driverless industrial truck	NS	Specifying safety requirements and verification for driverless industrial trucks
P019	ISO (2014)	NA	Robots and robotic devices	NS	Specifying safety requirements for personal care robots
P020	ISO (2018b)	NA	Earth-moving machinery	NS	Methodology to determine safety-related parts of the control system and performance requirements
P021	ISO (2017)	NA	Earth-moving machinery and mining	NS	Autonomous and semi-autonomous machine system safety
P022	ISO (2018a)	NA	Highly automated agricultural machines	NS	Defining principles for design for safety of highly automated agricultural machines

(*continued on next page*)

Table 6 (*continued*).

ID	References	Perspective	Domain	ML-technique	Objective
P023	IEC (2009)	NA	Railway applications	NS	Defining safety requirements for automated urban guided transport (AUGT)
P024	ISO (2019)	NA	Road vehicles	NS	Guiding on confirming safety of the intended functionality
P025	Koopman et al. (2019)	System engineer, safety engineer	Fully autonomous vehicle	NS	Setting a scope requirement for safety assurance case
P026	ISO/IEC (0000c)	AI experts	General	NS	Artificial intelligence — Concepts and terminology
P027	ISO (0000a)	AI experts	General	NS	Framework for AI systems using ML
P028	ISO/IEC (0000d)	AI experts	General	NS	Guidelines on risk management of AI
P029	ISO/IEC (0000b)	AI experts	General	NS	Guidelines to reduce bias in AI systems and AI aided decision making
P030	ISO (0000b)	AI experts	General	NS	Providing an overview of trustworthiness in artificial intelligence
P031	ISO/IEC (0000f)	AI experts	General	Neural Network	Providing guidelines on the assessment of the robustness of neural networks
P032	ISO/IEC (0000e)	AI experts	General	NS	Governance implications of the use of artificial intelligence by organizations
P033	ISO (2018)	System engineer, safety engineer	Road vehicles	NS	Provides vocabulary on road vehicle safety
P034	Salay et al. (2017)	ML expert, safety engineer, system engineer	General	NS	Discussing the impact of ML on the guidelines of ISO 26262.
P035	Yoshida (2019)	System engineer, safety engineer, ML expert	Autonomous car	NS	Describing a safety case approach to ensuring autonomous product safety
P036	Owens et al. (2008)	System engineer	Outer planet exploratory system	NS	Describing integrated safety-driven design methodology
P037	Bach et al. (2016)	System engineer, safety engineer	Autonomous car	NS	Specifying model-based scenarios to develop and test autonomous cars.
P038	Menzel et al. (2018)	System engineer, safety engineer	Automated vehicles	NS	
P039	Gu and Easwaran (2019)	ML experts	General	NS	Identifying training space and avoiding exploiting beyond that to ensure safety
P040	Chow et al. (2017)	ML experts	General	RL	Presenting efficient reinforcement learning algorithms for risk-constrained Markov decision processes
P041	Chen et al. (2020)	ML experts	Autonomous car	RL	Incorporating human decision-making model in RL to control AVs for safe operations.
P042	Rong and Luan (2020)	ML experts	Autonomous car	RL	Planning with safe policies for RL-based autonomous driving.
P043	Varshney (2016)	ML experts	General	NS	Defining the concept of safety, risk factors and safety strategies in case of ML.
P044	Kuper et al. (2018)	ML expert	General	DNN	Verification of DNN.
P045	Lee et al. (2020)	ML expert	General	DNN	Verification of DNN.
P046	Naseer et al. (2020)	ML expert	General	DNN	Verification of DNN.
P047	Sun et al. (2019)	ML expert	General	DNN	Verification of DNN.

(*continued on next page*)

Table 6 (*continued*).

ID	References	Perspective	Domain	ML-technique	Objective
P048	Harel-Canada et al. (2020)	ML expert	General	DNN	Discussing the right metric for verification of DNN
P049	Byun and Rayadurgam (2020)	ML expert	General	DNN	Proposing manifold-based test generation as a better metric for ML assurance than neuron coverage.
P050	Fazlyab et al. (2019)	ML expert	General	DNN	Verification and reachability analysis of DNN.
P051	Ivanov et al. (2020)	ML expert, system engineer	Autonomous car	DNN	Verifying safety of DNN-based autonomous cars
P052	Hoang et al. (2020)	ML expert	General	DNN	Analyzing resilience of DNN
P053	Koopman and Wagner (2018)	System engineer	Autonomous car	NS	Designing a framework for safety validation of autonomous cars
P054	Seshia et al. (2016)	ML expert	General	NS	Designing AI-based systems against verifiable requirements for better assurance.
P055	Ghosh et al. (2016)	Control system engineer	Autonomous driving, aircraft controller	NS	Diagnostic and repairing specification for hybrid systems
P056	Li et al. (2014)	Control system engineer	General	NS	Formalizing human-in-the-loop control systems
P057	Nilim and El Ghaoui (2005)	Control system engineer	General	NS	Handling robust control problem of Markov decision process.
P058	Fawzi et al. (2018)	ML expert	General	NN	Analyzing robustness of classifier against adversarial perturbation
P059	Seshia et al. (2018)	ML expert	General	DNN	Formally specifying DNN
P060	Sadigh and Kapoor (2016)	Control system engineer	Quadrotors, autonomous vehicles	NS	Achieving safe control with probabilistic signal temporal logic.
P061	Dutta et al. (2017)	ML expert	General	DNN	Analyzing output range for DNN
P062	Ehlers (2017)	ML expert	General	NN	Verification of NN
P063	Huang et al. (2017a)	ML expert	General	NN	Safety verification of NN
P064	Katz et al. (2017)	ML expert	General	NN	Verification of NN
P065	Nguyen et al. (2015)	ML expert	General	DNN	Analyzing adversarial attacks on DNN.
P066	Moosavi-Dezfooli et al. (2016)	ML expert	General	DNN	Analyzing adversarial attacks on DNN.
P067	Goodfellow et al. (2014)	ML expert	General	DNN	Analyzing adversarial attacks on DNN.
P068	Chakraborty et al. (2014)	Control system engineer	General	NS	Formal verification of control system
P069	Shoukry et al. (2017)	Control system engineer	General	NS	Formal verification of control system
P070	Tuncali et al. (2018b)	Control system engineer	General	DNN	Reasoning about safety of a NN-based system by a simulation-based approach
P071	Pei et al. (2017)	ML expert	General	NN	Testing of NN-based system
P072	Tian et al. (2018)	ML expert	Autonomous car	NN	Testing of NN-based system
P073	Wicker et al. (2018)	ML expert	General	NN	Testing of NN-based system
P074	Ma et al. (2018a)	ML expert	General	NN	Testing of NN-based system

(*continued on next page*)

Table 6 (*continued*).

ID	References	Perspective	Domain	ML-technique	Objective
P075	Sun et al. (2018a)	ML expert	General	NN	Testing of NN-based system
P076	Wang et al. (2019)	ML expert	General	NN	Testing of NN-based system
P077	Ma et al. (2018b)	ML expert	General	NN	Testing of NN-based system
P078	Sun et al. (2018b)	ML expert	General	NN	Testing of NN-based system
P079	Dreossi et al. (2019)	ML expert, system engineer	General	NS	Verification of system with ML-based components
P080	Tuncali et al. (2018a)	Control system engineer	Autonomous vehicle	NN	Testing of NN-based system
P081	Xiang and Johnson (2018)	Control system engineer	Control systems	NN	Testing and analyzing reachability of NN controllers
P082	Ivanov et al. (2019)	Control system engineer, ML expert	Control systems	NS	Safety verification of NN controller-based systems
P083	Akintunde et al. (2018)	ML expert, system engineer	Hybrid systems	NS	Analyzing reachability of neural agent-environment systems
P084	Dennis et al. (2016)	ML expert	Agent-based systems	NS	Practical verification of decision-making
P085	Alur (2011)	System engineer, ML expert	Robotics systems	NS	Formal verification of hybrid system
P086	Szegedy et al. (2013)	ML expert	General	NN	Discussing properties of NN
P087	Biggio et al. (2013)	ML expert	General	NS	Simulating evasion attack scenarios against ML algorithms at test time
P088	Papernot et al. (2017)	ML expert	General	NN	Analyzing attacks on DNN
P089	Bastani et al. (2016)	ML expert	General	NN	Measuring robustness of NN
P090	Grosse et al. (2016)	ML expert	Malware classification system	NN	Analyzing attacks on DNN
P091	Gu and Rigazio (2014)	ML expert	General	NN	Designing NN architecture robust to attacks
P092	Hazan et al. (2017)	ML expert	General	NN	Analyzing attacks on DNN
P093	Papernot et al. (2016)	ML expert	General	NN	Analyzing attacks on DNN
P094	Huang et al. (2017b)	ML expert	General	NN	Analyzing attacks on DNN
P095	Shaham et al. (2015)	ML expert	General	NN	Adversarial training to increase local stability of NN
P096	Carlini and Wagner (2017)	ML expert	General	NN	Evaluating robustness of NN
P097	Chen et al. (2017)	ML expert	General	NN	Analyzing attacks on DNN
P098	Kurakin et al. (2016)	ML expert	General	NN	Analyzing attacks on DNN
P099	Carlini et al. (2016)	ML expert	General	NN	Analyzing attacks on DNN
P100	Eykholt et al. (2018)	ML expert	General	NN	Analyzing attacks on DNN
P101	Palin et al. (2011)	Safety engineer	Automotive	NS	Safety assurance based on ISO 26262
P102	Gauerhof et al. (2018)	ML expert, systems engineer	Autonomous car	NS	Setting validation targets for ML-based automated driving
P103	Matsuno et al. (2019)	ML expert	General	NS	Safety assurance of ML
P104	Denney et al. (2015)	ML expert	General	NS	Safety assurance of ML-based system
P105	Picardi et al. (2019)	ML expert	Medical diagnosis system	DNN	Safety assurance of ML-based component

(*continued on next page*)

Table 6 (*continued*).

ID	References	Perspective	Domain	ML-technique	Objective
P106	Picardi and Habli (2019)	ML expert	Medical diagnosis system	DNN	Safety assurance of ML-based component
P107	Morton et al. (2018)	Systems engineer	General	NA	Testing safety-critical systems
P108	Leike et al. (2017)	ML expert	General	RL	Designing test suite for safety properties verification of RL-based agents
P109	Lwakatare et al. (2019)	Software engineers	General	NS	Discussing engineering challenges for ML-based systems.
P110	Varshney (2020)	ML expert, system engineer	General	NS	Discussing the necessity of the consensus on concepts like safety, trust, etc for ML-based safety-critical systems
P111	Sadigh et al. (2014)	Control system engineer	Autonomous control systems	NS	Modeling and verification of uncertain environment including human driver.
P112	Sadigh et al. (2016a)	Control system engineer	Autonomous control systems	NS	Modeling and verification of uncertain environment including human driver.

Table 7
Primary studies mapped to the challenges of each layer in the three-layered framework.

Layer	Challenges	Primary studies addressing the challenges
Layer-1: Problem Definition Layer	1. Understanding the problem domain.	P001, P004, P010
	2. Setting quantitative targets	P009
	3. Setting qualitative targets	P005, P008, P012, P013
	4. Requirements modeling techniques to address uncertainty	P011
	5. Explaining black box to users	P035, P039
	6. Declaring hidden customers	P002, P006, P010
Layer-2: Safety-driven modeling and analysis layer	1. Safety-driven system-level design	P036, P039, P040, P041, P042, P043
	2. Modeling uncertain environment	P056, P111, P112
	3. Scenario-based safe design and development	P037, P038
	4. Appropriate safety standards	P014, P015, P016, P017, P018, P019, P020, P021, P022, P023, P024, P025, P026, P027, P028, P029, P030, P031, P032, P033, P034, P035
Layer-3: Verified safety compliance layer	1. Formal method to verify the system	P059, P135
	2. Safety-modeling approaches for ML-based components	P059, P135
	3. Quantitative verification	P054, P055, P058, P065, P066, P067, P068, P069, P070, P071, P072, P073, P074, P075, P076, P077, P078, P079, P080, P081, P082, P083, P045, P047, P049, P050
	4. Rigorous run-time testing	P053, P107, P051, P052
	5. Robustness against adversarial attacks	P086, P087, P088, P089, P090, P091, P092, P093, P094, P095, P096, P096, P097, P098, P099, P100, P046, P065, P066, P067
	6. Demonstrating assurance cases	P101, P102, P103, P104, P105, P106

mapped to the design phase of ML-based component, rather than its development phase. All the safety-related standards are mapped to system-level development as the standards usually provide guidelines to the overall development process of the safety-critical systems. As shown in Fig. 5, there has been a recent surge in the research on verification and validation of safety-critical systems with ML-based components since 2016. However, more effort needs to be paid to conceptualize and analyze the rich problem space of AI systems in the early phases such as requirements engineering and design. It is very important to have a systematic start of the engineering process as it often

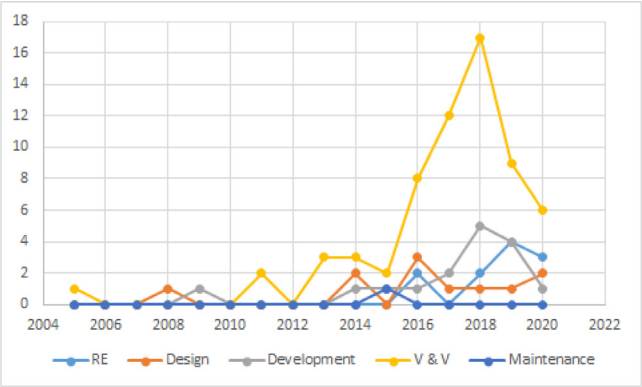

Fig. 5. Trend of research on safety approaches from an engineering perspective in the last 15 years.

plays a key role in successful product development. Moreover, not enough attention is paid to the seamless maintenance and evolution of complex AI systems as it can be seen in both Figs. 4 and 5.

6.3. RQ3: Research gap analysis

The most significant concern is that two different communities or engineering disciplines are responsible for ensuring safety of intelligent systems. Safety as argued by many researchers is an emergent property of a system. The source of hazard at a system level could be control flaws, inadequate control action, and inadequate control execution, etc. Artifacts and metrics for the safety analysis at the system level, such as safety goal model, safety argument, and safety integrity level, etc. have already been identified. However, when intelligence is introduced in the system through an ML component mainly designed by AI experts, the identified source of hazards are different in terms of complexity and level of abstraction. An ML component can perform incorrectly because of the wrong choice of training data (supervised learning) or wrong choice of policy, unsafe exploration, delayed rewards, etc. (unsupervised learning). There is no common baseline of the metrics and artifacts to ensure or explain the safety constraint from an ML point of view. In the same direction, Varshney discussed a much-needed future technical agenda on defining how trust, reliability, robustness, etc. are traced to safety (Varshney, 2020). There is a significant gap between the two levels of safety analysis at the system level and the component level whenever any intelligence is introduced by using ML component (Fig. 6). The open research questions are:

(1) How to bridge the safety analysis gap from a safety viewpoint?

(2) How to enhance traceability of these artifacts from the system level down to the component level?

(3) What is the formal taxonomy for safety analysis in the case of ML components?

(4) Is there any standard specially designed to assess the level of safety and the acceptable range of uncertainty of ML-based components in complex AI systems?

The gap between research and practice: Koopman explained that there is a significant contrast between the safety principles in research and practice in automotive industry (Koopman, 2018a). He argued that although ISO 26262 looks promising as a standard for the level of safety of automotive systems, uncertainty in the operating environment still poses some threat to complete deployment. Salay made a notable contribution to update ISO 26262 to accommodate technologies such as ML (Salay et al., 2017). Lawakate et al. conducted an empirical investigation to deduce a taxonomy of SE challenges for several domains that use ML-based components heavily (Lwakatare et al., 2019). The authors emphasized that the challenges of a seamless engineering process that includes the development and evolution of ML-based components are very significant.

6.4. RQ4: Future scope:

6.4.1. Addressing multi-disciplinary challenge: Collective intelligence

The recent accidents on the autonomous systems were mostly linked to insufficient training (inadequate dataset) or choosing the wrong level of automation. The root cause was insufficient coordination between ML experts and safety engineers at an early phase of systems engineering. Requirement specification should describe data requirements, values consensus (from multiple participants), the reaction of a system to a fault, etc. This is a multidisciplinary challenge where collective intelligence can be a great asset. To ensure safety, many different domain areas need to be coordinated as shown in the framework. Data scientists, HCI experts, safety engineers, software engineers, etc. all need to collaborate to ensure safety. Instead of relying on the knowledge of individuals, we should use the collection as an emergent intelligence to solve engineering problems. We can rely on all to work together, collaborate, and share individual knowledge to achieve a goal. In a nutshell, we can have a paradigm shift from traditional safety engineering to a broader concept of collective knowledge engineering.

6.4.2. Enhanced traceability of the artifacts across the layers of the integrated framework

Due to the lack of an integrated framework, it is unclear how we can trace safety-driven design of AI systems. Most of the research work focusing on safety-driven ML do not specify clearly how their design/ learning decisions relate to the system level safety-related concepts. Traditional understanding of forward and backward traceability has less impact in the days of AI. Therefore, we define horizontal and vertical traceability of safety-driven design of AI systems.

Horizontal Traceability: traces of safety engineering process and design decisions along the same layer at system-level and ML-based component-level involving multiple participants.

Vertical Traceability: traces of safety analysis across the layers.

Instead of relying on the tacit knowledge of the experts, we recommend keeping a record of all design decisions and the rationales behind them. Therefore, we need to specify ML related categories of risks, the way those risks are adding on to the system-level risks. Thereafter, the experts can mitigate each risk with safety constraints and record them in the artifacts.

7. Threats to validity

In this section we discuss the known threats to the validity of our literature survey. We also explain how we mitigated the threats. Moreover, we summarize the limitation of our study. As discussed in Zhou et al. (2016), identified threats to four types of validity (internal, external, construct and conclusion) can be mapped to the following phases of the literature survey.

1. Planning phase:

In this phase, we set the valid research questions which were evaluated later by the graduate researchers working on AI systems engineering. We documented the rationale behind each of the research questions. We followed a stepwise process for the search method with explicit actions and outcomes. The complete list of venues or databases and search strings are mentioned in Sections 4.2 and 4.3, so that the SLR can be replicated in future.

Table 8
State-of-the art safety approaches for complex intelligent systems with respect to software engineering process.

Level	SE Phases				
	Requirements Engineering	Design	Development	V & V	Maintenance
System level	P007, P010, P011, P017, P037, P038	P036, P054, P055, P111, P112	P006, P007, P014, P015, P016, P018, P019, P020, P021, P022, P023, P024 P025, P026, P027, P028, P029, P030, P031, P032, P033, P035, P037, P038	P035, P051, P053, P060, P071, P072, P079, P080, P081, P082, P083, P084, P085, P101, P107, P108	P104
ML component level	P004, P005, P008, P009, P012, P013	P039, P040, P041, P042, P043, P044, P056, P067, P091	–	P044, P045, P046, P047, P048, P049, P050, P052, P057, P058, P059, P061, P062, P063, P064, P065, P066, P067, P069, P070, P073, P074, P075, P076, P077, P078, P086, P087, P088, P089, P090, P091, P092, P093, P095, P097, P098, P099, P100, P101, P102, P103, P105, P106	–

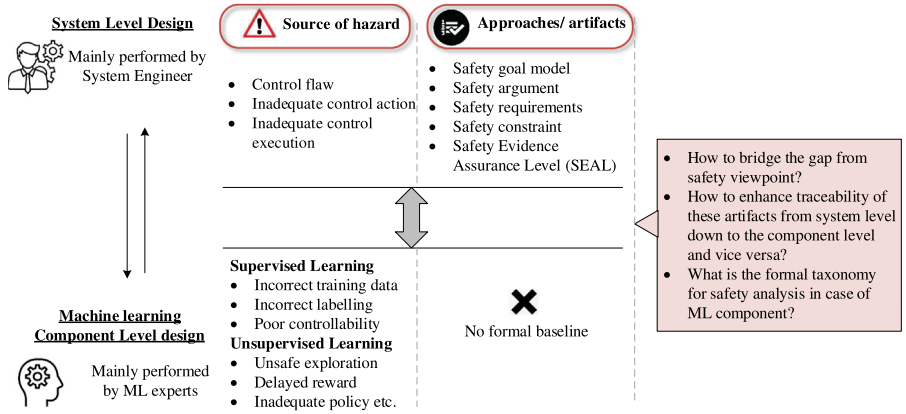

Fig. 6. Gap analysis of the state-of-the art safety approaches of complex intelligent systems.

The finalized list of primary studies after applying the inclusion and exclusion criteria are rechecked at least twice before moving to the next phase.

2. Conducting phase:

In order to mitigate the threat of incomplete research information or inaccessible full version of papers, we have contacted the relevant authors whenever needed. We mitigated the publication bias by excluding the gray literature from the study. During this phase, each of the paper has been cross-checked for their completeness. Along with the title and abstract, the introduction of each paper was carefully read to confirm that the perspective the work is not misunderstood. The quality of the papers is evaluated quantitatively to mitigate the threat of subjective quality assessment. Threat of duplication was mitigated by carefully choosing the full paper version (if available) over a smaller and earlier version of the same work.

3. Reporting phase:

In this phase, we report our analysis based on the research questions after analyzing 112 primary studies. These many studies are good enough to mitigate the threat of low generalizability of primary studies. We have covered safety-related papers written from various perspectives (software engineer, ML experts, safety engineer, etc.) and applied to a diverse domain, to increase the generalizability of our final report.

Limitations:

Our study is primarily based on the proposed three-layered conceptual framework to engineer AI systems. This framework is designed based on our knowledge on the relevant area of research. It helps us to analyze the state-of-the-art safety approaches from the perspective of multiple stakeholders along each phase of the system engineering. However, an end-to-end evaluation of the effectiveness of the proposed framework is out of the scope of this paper. The survey covers primary studies mainly from the software engineering process perspective. However, there are other relevant studies on AI safety can be found from the discipline of statistics, human–computer interaction, etc. Those studies are not discussed in this literature review.

8. Conclusion

The use of ML techniques to impart intelligence creates many challenges to ensuring safety. In this paper, we summarized the current state-of-the-art research contributions in this area. We explained a three-layered conceptual framework that can help visualize the stakeholders and their contributions to engineering a complex intelligent system from a safety perspective. We analyzed the gap in the current research that should be addressed. We also described how this three-layered framework can help enhance the traceability of safety-driven design across all the layers in the future. We believe that it is worth exploring how the artifacts move across the layers in different forms to facilitate the safety analysis. Instead of relying on tacit design knowledge (or tacit machine learning knowledge of AI experts), it is helpful to document all the rationales behind each decision to eventually gain the trust of the stakeholders. For future work, we plan to design a proper stepwise methodology to guide multiple disciplines to work together for safety analysis and verification of AI systems.

CRediT authorship contribution statement

Sangeeta Dey: Conceptualization, Writing - original draft, Data curation, Investigation. **Seok-Won Lee:** Conceptualization, Investigation, Writing - review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1F1A1075605).

References

Agrell, C., Eldevik, S., Hafver, A., Pedersen, F.B., Stensrud, E., Huseby, A., 2018. Pitfalls of Machine Learning for Tail Events in High Risk Environments. Taylor & Francis, http://dx.doi.org/10.1201/9781351174664-381.

Akintunde, M., Lomuscio, A., Maganti, L., Pirovano, E., 2018. Reachability analysis for neural agent-environment systems. In: KR. pp. 184–193.

Alur, R., 2011. Formal verification of hybrid systems. In: Proceedings of the Ninth ACM International Conference on Embedded Software, pp. 273–278.

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

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D., 2016. Concrete problems in AI safety. ArXiv preprint arXiv:1606.06565.

Arpteg, A., Brinne, B., Crnkovic-Friis, L., Bosch, J., 2018. Software engineering challenges of deep learning. In: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications. SEAA. IEEE, pp. 50–59.

Bach, J., Otten, S., Sax, E., 2016. Model based scenario specification for development and test of automated driving functions. In: 2016 IEEE Intelligent Vehicles Symposium. IV. IEEE, pp. 1149–1155.

Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A., Criminisi, A., 2016. Measuring neural net robustness with constraints. In: Advances in Neural Information Processing Systems. pp. 2613–2621.

Belani, H., Vukovic, M., Car, Ž., 2019. Requirements engineering challenges in building AI-based complex systems. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops. REW. IEEE, pp. 252–255.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F., 2013. Evasion attacks against machine learning at test time. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 387–402.

Borg, M., Englund, C., Wnuk, K., Duran, B., Levandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., Törnqvist, J., 2018. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. ArXiv preprint arXiv:1812.05389.

Bosch, J., Olsson, H.H., Crnkovic, I., 2018. It takes three to tango: Requirement, outcome/data, and AI driven development. In: SiBW. pp. 177–192.

Brundage, M., 2015. Taking superintelligence seriously: Superintelligence: Paths, dangers, strategies by Nick Bostrom (Oxford University Press, 2014). Futures 72, 32–35.

Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., et al., 2020. Toward trustworthy AI development: mechanisms for supporting verifiable claims. ArXiv preprint arXiv:2004.07213.

Byun, T., Rayadurgam, S., 2020. Manifold for machine learning assurance. ArXiv preprint arXiv:2002.03147.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W., 2016. Hidden voice commands. In: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 513–530.

Carlini, N., Wagner, D., 2017. Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy. Sp. IEEE, pp. 39–57.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D., 2018. Adversarial attacks and defences: A survey. ArXiv preprint arXiv:1810.00069.

Chakraborty, S., Fremont, D.J., Meel, K.S., Seshia, S.A., Vardi, M.Y., 2014. Distribution-aware sampling and weighted model counting for SAT. ArXiv preprint arXiv:1404.2984.

Chen, D., Jiang, L., Wang, Y., Li, Z., 2020. Autonomous driving using safe reinforcement learning by incorporating a regret-based human lane-changing decision model. In: 2020 American Control Conference. ACC. IEEE, pp. 4355–4361.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.-J., 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26.

Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M., 2017. Risk-constrained reinforcement learning with percentile risk criteria. J. Mach. Learn. Res. 18 (1), 6070–6120.

Cihon, P., 2019. Standards for AI Governance: International Standards to Enable Global Coordination in AI Research & Development. Future of Humanity Institute. University of Oxford.

Davis, E., 2015. Ethical guidelines for a superintelligence. Artificial Intelligence 220, 121–124.

Denney, E., Pai, G., Habli, I., 2015. Dynamic safety cases for through-life safety assurance. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. 2, IEEE, pp. 587–590.

Dennis, L.A., Fisher, M., Lincoln, N.K., Lisitsa, A., Veres, S.M., 2016. Practical verification of decision-making in agent-based autonomous systems. Autom. Softw. Eng. 23 (3), 305–359.

Dewey, D., Russell, S., Tegmark, M., et al., 2015. A survey of research questions for robust and beneficial AI. Future of life institute.

Dey, S., Lee, S.-W., 2021. Replication Package for the Multilayered Review of Safety Approaches for Machine Learning-based Systems in the Days of AI (Version 1.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4576996.

Dreossi, T., Donzé, A., Seshia, S.A., 2019. Compositional falsification of cyber-physical systems with machine learning components. J. Automat. Reason. 63 (4), 1031–1053.

Dutta, S., Jha, S., Sanakaranarayanan, S., Tiwari, A., 2017. Output range analysis for deep neural networks. ArXiv preprint arXiv:1709.09130.

Ehlers, R., 2017. Formal verification of piece-wise linear feed-forward neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Springer, pp. 269–286.

Everitt, T., Lea, G., Hutter, M., 2018. AGI Safety literature review. ArXiv preprint arXiv:1805.01109.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D., 2018. Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1625–1634.

Fawzi, A., Fawzi, O., Frossard, P., 2018. Analysis of classifiers' robustness to adversarial perturbations. Mach. Learn. 107 (3), 481–508.

Fazlyab, M., Morari, M., Pappas, G.J., 2019. Probabilistic verification and reachability analysis of neural networks via semidefinite programming. In: 2019 IEEE 58th Conference on Decision and Control. CDC. IEEE, pp. 2726–2731.

Firesmith, D., 2004. Engineering safety requirements, safety constraints, and safety-critical requirements. J. Object Technol. 3 (3), 27–42.

Garcıa, J., Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. 16 (1), 1437–1480.

Gauerhof, L., Munk, P., Burton, S., 2018. Structuring validation targets of a machine learning function applied to automated driving. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 45–58.

Ghosh, S., Sadigh, D., Nuzzo, P., Raman, V., Donzé, A., Sangiovanni-Vincentelli, A.L., Sastry, S.S., Seshia, S.A., 2016. Diagnosis and repair for synthesis from signal temporal logic specifications. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, pp. 31–40.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. ArXiv preprint arXiv:1412.6572.

Google White Paper, 2019. Perspectives on Issues in AI Governance.

Greenblatt, N.A., 2016. Self-driving cars and the law. IEEE Spectrum 53 (2), 46–51.

Grosse, K., Papernot, N., Manoharan, P., Backes, M., McDaniel, P., 2016. Adversarial perturbations against deep neural networks for malware classification. ArXiv preprint arXiv:1606.04435.

Gu, X., Easwaran, A., 2019. Towards safe machine learning for CPS: infer uncertainty from training data. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, pp. 249–258.

Gu, S., Rigazio, L., 2014. Towards deep neural network architectures robust to adversarial examples. ArXiv preprint arXiv:1412.5068.

Guiochet, J., Machin, M., Waeselynck, H., 2017. Safety-critical advanced robots: A survey. Robot. Auton. Syst. 94, 43–52.

Guo, J., Kurup, U., Shah, M., 2019. Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving. IEEE Trans. Intell. Transp. Syst..

Harel-Canada, F., Wang, L., Gulzar, M.A., Gu, Q., Kim, M., 2020. Is neuron coverage a meaningful measure for testing deep neural networks? In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 851–862.

Hazan, T., Papandreou, G., Tarlow, D., 2017. Adversarial Perturbations of Deep Neural Networks. MIT Press.

Hoang, L.-H., Hanif, M.A., Shafique, M., 2020. Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation. In: 2020 Design, Automation & Test in Europe Conference & Exhibition. DATE. IEEE, pp. 1241–1246.

Horkoff, J., 2019. Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27th International Requirements Engineering Conference. RE. IEEE, pp. 386–391.

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

Huang, X., Kwiatkowska, M., Wang, S., Wu, M., 2017a. Safety verification of deep neural networks. In: International Conference on Computer Aided Verification. Springer, pp. 3–29.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., Abbeel, P., 2017b. Adversarial attacks on neural network policies. ArXiv preprint arXiv:1702.02284.

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

Ishikawa, F., Matsuno, Y., 2020. Evidence-driven requirements engineering for uncertainty of machine learning-based systems. In: 2020 IEEE 28th International Requirements Engineering Conference. RE. IEEE, pp. 346–351.

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

ISO, In progress.b ISO/IEC CD 23053.2- Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML).

ISO, In progress.e ISO/IEC TR 24028:2020 -Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence.

ISO, 2014. ISO 13482:2014- robots and robotic devices — Safety requirements for personal care robots.

ISO, 2017. ISO 17757:2017- Earth-moving machinery and mining — Autonomous and semi-autonomous machine system safety.

ISO, 2018. ISO 26262:2018-Road Vehicles- Vehicle Safety. ISO.

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

ISO, 2018b. ISO 19014-1:2018- Earth-moving machinery — Functional safety — Part 1: Methodology to determine safety-related parts of the control system and performance requirements.

ISO, 2019. ISO/PAS 21448:2019- Road vehicles – Safety of the intended functionality.

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

ISO/IEC, In progress.d ISO/IEC AWI TR 24027- Information technology — Artificial Intelligence (AI) — Bias in AI systems and AI aided decision making.

ISO/IEC, In progress.a ISO/IEC CD 22989.2-Artificial intelligence — Concepts and terminology.

ISO/IEC, In progress.c ISO/IEC CD 23894- Information Technology — Artificial Intelligence — Risk Management.

ISO/IEC, In progress.g ISO/IEC CD 38507-Information technology — Governance of IT — Governance implications of the use of artificial intelligence by organizations.

ISO/IEC, In progress.f ISO/IEC DTR 24029-1 Artificial Intelligence (AI) — Assessment of the robustness of neural networks.

ISO/IEC, 2010a. IEC 61508-Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems.

Ivanov, R., Carpenter, T.J., Weimer, J., Alur, R., Pappas, G.J., Lee, I., 2020. Case study: verifying the safety of an autonomous racing car with a neural network controller. In: Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control, pp. 1–7.

Ivanov, R., Weimer, J., Alur, R., Pappas, G.J., Lee, I., 2019. Verisig: verifying safety properties of hybrid systems with neural network controllers. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, pp. 169–178.

Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J., 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In: International Conference on Computer Aided Verification. Springer, pp. 97–117.

Köhl, M.A., Baum, K., Langer, M., Oster, D., Speith, T., Bohlender, D., 2019. Explainability as a non-functional requirement. In: 2019 IEEE 27th International Requirements Engineering Conference. RE. IEEE, pp. 363–368.

Koopman, P., 2018a. Practical experience report: Automotive safety practices vs. accepted principles. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 3–11.

Koopman, P., 2018b. The heavy tail safety ceiling. In: Automated and Connected Vehicle Systems Testing Symposium.

Koopman, P., Ferrell, U., Fratrik, F., Wagner, M., 2019. A safety standard approach for fully autonomous vehicles. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 326–332.

Koopman, P., Wagner, M., 2017. Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intell. Transp. Syst. Mag. 9 (1), 90–96.

Koopman, P., Wagner, M., 2018. Toward a Framework for Highly Automated Vehicle Safety Validation. SAE Technical Paper.

Kuper, L., Katz, G., Gottschlich, J., Julian, K., Barrett, C., Kochenderfer, M., 2018. Toward scalable verification for safety-critical deep networks. ArXiv preprint arXiv:1801.05950.

Kurakin, A., Goodfellow, I., Bengio, S., 2016. Adversarial examples in the physical world. ArXiv preprint arXiv:1607.02533.

Lee, S., Cha, S., Lee, D., Oh, H., 2020. Effective white-box testing of deep neural networks with adaptive neuron-selection strategy. In: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 165–176.

Leike, J., Martic, M., Krakovna, V., Ortega, P.A., Everitt, T., Lefrancq, A., Orseau, L., Legg, S., 2017. AI Safety gridworlds. ArXiv preprint arXiv:1711.09883.

Leveson, N., 2004a. A new accident model for engineering safer systems. Saf. Sci. 42 (4), 237–270.

Leveson, N.G., 2004b. A systems-theoretic approach to safety in software-intensive systems. IEEE Trans. Dependable Secure Comput. 1 (1), 66–86.

Leveson, N.G., 2016. Engineering a Safer World: Systems Thinking Applied to Safety. The MIT Press.

Li, W., Dworkin, L., Seshia, S.A., 2011. Mining assumptions for synthesis. In: Ninth ACM/IEEE International Conference on Formal Methods and Models for Codesign. MEMPCODE2011. IEEE, pp. 43–50.

Li, W., Sadigh, D., Sastry, S.S., Seshia, S.A., 2014. Synthesis for human-in-the-loop control systems. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, pp. 470–484.

Littlewood, B., Wright, D., 2007. The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example. IEEE Trans. Softw. Eng. 33 (5), 347–365.

Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M., 2019. Formal specification and verification of autonomous robotic systems: A survey. ACM Comput. Surv. 52 (5), 1–41.

Luo, Y., Yu, Y., Jin, Z., Zhao, H., 2019. Environment-centric safety requirements for autonomous unmanned systems. In: 2019 IEEE 27th International Requirements Engineering Conference. RE. IEEE, pp. 410–415.

Lwakatare, L.E., Raj, A., Bosch, J., Olsson, H.H., Crnkovic, I., 2019. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In: International Conference on Agile Software Development. Springer, Cham, pp. 227–243.

Lwakatare, L.E., Raj, A., Crnkovic, I., Bosch, J., Olsson, H.H., 2020. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. Inf. Softw. Technol. 127, 106368.

Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M., Li, B., Chen, C., Su, T., Li, L., Liu, Y., et al., 2018a. Deepgauge: Multi-granularity testing criteria for deep learning systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 120–131.

Ma, L., Zhang, F., Sun, J., Xue, M., Li, B., Juefei-Xu, F., Xie, C., Li, L., Liu, Y., Zhao, J., et al., 2018b. Deepmutation: Mutation testing of deep learning systems. In: 2018 IEEE 29th International Symposium on Software Reliability Engineering. ISSRE. IEEE, pp. 100–111.

Mallah, R., 2017. The landscape of AI safety and beneficence research. Input for brainstorming at beneficial AI 2017. In: Beneficial AI 2017.

Martínez-Fernández, S., Franch, X., Jedlitschka, A., Oriol, M., Trendowicz, A., 2020. Research directions for developing and operating artificial intelligence models in trustworthy autonomous systems. ArXiv preprint arXiv:2003.05434.

Matsuno, Y., Ishikawa, F., Tokumoto, S., 2019. Tackling uncertainty in safety assurance for machine learning: Continuous argument engineering with attributed tests. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 398–404.

Menzel, T., Bagschik, G., Maurer, M., 2018. Scenarios for development, test and validation of automated vehicles. In: 2018 IEEE Intelligent Vehicles Symposium. IV. IEEE, pp. 1821–1827.

Mikolov, T., Joulin, A., Baroni, M., 2016. A roadmap towards machine intelligence. In: International Conference on Intelligent Text Processing and Computational Linguistics. Springer, pp. 29–61.

Monkhouse, H., Habli, I., McDermid, J., Khastgir, S., Dhadyalla, G., 2017. Why functional safety experts worry about automotive systems having increasing autonomy. In: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, pp. 1–6.

Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P., 2016. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582.

Morton, J., Wheeler, T.A., Kochenderfer, M.J., 2018. Closed-loop policies for operational tests of safety-critical systems. IEEE Trans. Intell. Veh. 3 (3), 317–328.

Nakamichi, K., Ohashi, K., Namba, I., Yamamoto, R., Aoyama, M., Joeckel, L., Siebert, J., Heidrich, J., 2020. Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In: 2020 IEEE 28th International Requirements Engineering Conference. RE. IEEE, pp. 260–270.

Nascimento, A.M., Vismari, L.F., Molina, C.B.S.T., Cugnasca, P.S., Camargo, J.B., de Almeida, J.R., Inam, R., Fersman, E., Marquezini, M.V., Hata, A.Y., 2019. A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. IEEE Trans. Intell. Transp. Syst.

Naseer, M., Minhas, M.F., Khalid, F., Hanif, M.A., Hasan, O., Shafique, M., 2020. Fannet: formal analysis of noise tolerance, training bias and input sensitivity in neural networks. In: 2020 Design, Automation & Test in Europe Conference & Exhibition. DATE. IEEE, pp. 666–669.

Nguyen, A., Yosinski, J., Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 427–436.

Nguyen-Duc, A., Abrahamsson, P., 2020. Continuous experimentation on artificial intelligence software: a research agenda. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1513–1516.

Nilim, A., El Ghaoui, L., 2005. Robust control of Markov decision processes with uncertain transition matrices. Oper. Res. 53 (5), 780–798.

Owens, B.D., Herring, M.S., Dulac, N., Leveson, N.G., Ingham, M.D., Weiss, K.A., 2008. Application of a safety-driven design methodology to an outer planet exploration mission. In: 2008 IEEE Aerospace Conference. IEEE, pp. 1–24.

Ozdag, M., 2018. Adversarial attacks and defenses against deep neural networks: a survey. Procedia Comput. Sci. 140, 152–161.

Ozlati, S., Yampolskiy, R., 2017. The formalization of AI risk management and safety standards. In: Workshops At the Thirty-First AAAI Conference on Artificial Intelligence.

Palin, R., Ward, D., Habli, I., Rivett, R., 2011. ISO 26262 Safety Cases: Compliance and Assurance. IET.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A., 2017. Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A., 2016. The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy, EuroS&P. IEEE, pp. 372–387.

Pei, K., Cao, Y., Yang, J., Jana, S., 2017. Deepxplore: Automated whitebox testing of deep learning systems. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp. 1–18.

Picardi, C., Habli, I., 2019. Perspectives on assurance case development for retinal disease diagnosis using deep learning. In: Conference on Artificial Intelligence in Medicine in Europe. Springer, pp. 365–370.

Picardi, C., Hawkins, R., Paterson, C., Habli, I., 2019. A pattern for arguing the assurance of machine learning in medical diagnosis systems. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 165–179.

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

Rong, J., Luan, N., 2020. Safe reinforcement learning with policy-guided planning for autonomous driving. In: 2020 IEEE International Conference on Mechatronics and Automation. ICMA. IEEE, pp. 320–326.

Russell, S., Dewey, D., Tegmark, M., 2015. Research priorities for robust and beneficial artificial intelligence. AI Mag. 36 (4), 105–114.

Sadigh, D., Driggs-Campbell, K., Puggelli, A., Li, W., Shia, V., Bajcsy, R., Sangiovanni-Vincentelli, A.L., Sastry, S.S., Seshia, S.A., 2014. Data-driven probabilistic modeling and verification of human driver behavior. In: AAAI Spring Symposium-Technical Report, pp. 56–61.

Sadigh, D., Kapoor, A., 2016. Safe control under uncertainty with probabilistic signal temporal logic.

Sadigh, D., Sastry, S.S., Seshia, S.A., Dragan, A., 2016a. Information gathering actions over human internal state. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS. IEEE, pp. 66–73.

Sadigh, D., Sastry, S., Seshia, S.A., Dragan, A.D., 2016b. Planning for autonomous cars that leverage effects on human actions. In: Robotics: Science and Systems, vol. 2. Ann Arbor, MI, USA.

Salay, R., Queiroz, R., Czarnecki, K., 2017. An analysis of ISO 26262: Using machine learning safely in automotive software. ArXiv preprint arXiv:1709.02435.

Schumann, J., Gupta, P., Liu, Y., 2010. Application of neural networks in high assurance systems: A survey. In: Applications of Neural Networks in High Assurance Systems. Springer, pp. 1–19.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., Dennison, D., 2015. Hidden technical debt in machine learning systems. In: Advances in Neural Information Processing Systems. pp. 2503–2511.

Serban, A., van der Blom, K., Hoos, H., Visser, J., 2020. Adoption and Effects of software engineering best practices in machine learning, in: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, pp. 1–12.

Seshia, S.A., Desai, A., Dreossi, T., Fremont, D.J., Ghosh, S., Kim, E., Shivakumar, S., Vazquez-Chanlatte, M., Yue, X., 2018. Formal specification for deep neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Springer, pp. 20–34.

Seshia, S.A., Sadigh, D., Sastry, S.S., 2016. Towards verified artificial intelligence. ArXiv preprint arXiv:1606.08514.

Shaham, U., Yamada, Y., Negahban, S., 2015. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. ArXiv preprint arXiv:1511.05432.

Shoukry, Y., Nuzzo, P., Sangiovanni-Vincentelli, A.L., Seshia, S.A., Pappas, G.J., Tabuada, P., 2017. SMC: Satisfiability modulo convex optimization. In: Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, pp. 19–28.

Singla, K., Bose, J., Naik, C., 2018. Analysis of software engineering for agile machine learning projects. In: 2018 15th IEEE India Council International Conference, INDICON. IEEE, pp. 1–5.

Soares, N., Fallenstein, B., 2014. Aligning Superintelligence with Human Interests: A Technical Research Agenda. Machine Intelligence Research Institute (MIRI) Technical Report 8, Citeseer.

Stringfellow, M.V., Leveson, N.G., Owens, B.D., 2010a. Safety-driven design for software-intensive aerospace and automotive systems. Proc. IEEE 98 (4), 515–525.

Stringfellow, M.V., Leveson, N.G., Owens, B.D., 2010b. Safety-driven design for software-intensive aerospace and automotive systems. Proc. IEEE 98 (4), 515–525.

Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., Ashmore, R., 2018a. Testing deep neural networks. ArXiv preprint arXiv:1803.04792.

Sun, X., Khedr, H., Shoukry, Y., 2019. Formal verification of neural network controlled autonomous systems. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, pp. 147–156.

Sun, Y., Wu, M., Ruan, W., Huang, X., Kwiatkowska, M., Kroening, D., 2018b. Concolic testing for deep neural networks. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 109–119.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. ArXiv preprint arXiv:1312.6199.

Tian, Y., Pei, K., Jana, S., Ray, B., 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering, pp. 303–314.

Tuncali, C.E., Fainekos, G., Ito, H., Kapinski, J., 2018a. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In: 2018 IEEE Intelligent Vehicles Symposium. IV. IEEE, pp. 1555–1562.

Tuncali, C.E., Kapinski, J., Ito, H., Deshmukh, J.V., 2018b. Reasoning about safety of learning-enabled components in autonomous cyber-physical systems. In: Proceedings of the 55th Annual Design Automation Conference, pp. 1–6.

Varshney, K.R., 2016. Engineering safety in machine learning. In: 2016 Information Theory and Applications Workshop, ITA. IEEE, pp. 1–5.

Varshney, K.R., 2020. On mismatched detection and safe, trustworthy machine learning. In: 2020 54th Annual Conference on Information Sciences and Systems. CISS. IEEE, pp. 1–4.

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

Wan, Z., Xia, X., Lo, D., Murphy, G.C., 2019. How does machine learning change software development practices? IEEE Trans. Softw. Eng.

Wang, J., Dong, G., Sun, J., Wang, X., Zhang, P., 2019. Adversarial sample detection for deep neural network through model mutation testing. In: 2019 IEEE/ACM 41st International Conference on Software Engineering. ICSE. IEEE, pp. 1245–1256.

Wicker, M., Huang, X., Kwiatkowska, M., 2018. Feature-guided black-box safety testing of deep neural networks. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, pp. 408–426.

Winfield, A.F., Blum, C., Liu, W., 2014. Towards an ethical robot: internal models, consequences and ethical action selection. In: Conference Towards Autonomous Robotic Systems. Springer, pp. 85–96.

Wohlin, C., 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10.

Xiang, W., Johnson, T.T., 2018. Reachability analysis and safety verification for neural network control systems. ArXiv preprint arXiv:1805.09944.

Xiang, W., Musau, P., Wild, A.A., Lopez, D.M., Hamilton, N., Yang, X., Rosenfeld, J., Johnson, T.T., 2018. Verification for machine learning, autonomy, and neural networks survey. ArXiv preprint arXiv:1810.01989.

Xu, H., Ma, Y., Liu, H.-C., Deb, D., Liu, H., Tang, J.-L., Jain, A.K., 2020. Adversarial attacks and defenses in images, graphs and text: A review. Int. J. Autom. Comput. 17 (2), 151–178.

Yoshida, J., 2019. UL 4600 draft puts safety onus on AV hopefuls. https://www.eetimes.com/ul-4600-draft-puts-safety-onus-on-av-hopefuls/#.

Yoshuda, J., 2020. Multiple standards to emerge in 2020 for AI-driven vehicles. (Accessed 27 November 2020).

Yurtsever, E., Lambert, J., Carballo, A., Takeda, K., 2020. A survey of autonomous driving: Common practices and emerging technologies. IEEE Access 8, 58443–58469.

Zhang, J.M., Harman, M., Ma, L., Liu, Y., 2020. Machine learning testing: Survey, landscapes and horizons. IEEE Trans. Softw. Eng.

Zhou, X., Jin, Y., Zhang, H., Li, S., Huang, X., 2016. A map of threats to validity of systematic literature reviews in software engineering. In: 2016 23rd Asia-Pacific Software Engineering Conference. APSEC. IEEE, pp. 153–160.

Sangeeta Dey is a Ph.D. student of Department of Artificial Intelligence at Ajou University in Republic of Korea. Her research interests include safety of machine learning-based systems, reasoning about uncertainty, and software engineering.

Dr. Seok-Won Lee is a Full Professor and Chair of the Dept. of Software and Computer Engineering and Dept. of Applied Artifical Intelligence, and Head of Graduate School of Software at Ajou University in Republic of Korea. He was a faculty member at the University of Texas at San Antonio and University of North Carolina at Charlotte in USA. He also worked at Science Applications International Corporation (SAIC) and IBM T.J. Watson Research Center as a senior research scientist. His areas of specialization include software engineering, knowledge acquisition, machine learning, and information assurance. He has published more than 180 peer reviewed articles. He is a senior member of IEEE, ACM and AAAI.